Summer 2019

Emulator for Water Erosion Prediction Project computer model using Gaussian Processes with functional inputs

Gulzina Kuttubekova

Emulator for Water Erosin Prediction Project computer model using Gaussian Processes with functional inputs

by

Gulzina Kuttubekova

A Creative Component submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Statistics

Program of Study Committee:
Jarad Niemi, Major Professor
Somak Dutta
Max Morris

Iowa State University
Ames, Iowa
2019

DEDICATION

I would like to dedicate this thesis to my husband Zamir, mom Altynai and dad Kuttubek without whose support I would not have been able to complete this work. Thanks for their loving guidance during the writing of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Jarad Niemi for his guidance, patience, and support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Max Morris, Dr.Petruţa Caragea and Dr. Somak Dutta. Additional thanks to my colleagues at C-CHANGE group for their help and support.

The main objective of this research is to build an emulator (i.e., surrogate) for the (Water Erosion Prediction Project) WEPP computer model, a soil erosion prediction technology used by the United States Department of Agriculture (USDA). The WEPP is a continuous simulation computer program that predicts soil loss and sediment deposition by considering various functional, quantitative and categorical inputs.

The emulator is built using Gaussian processes (GP) with both scalar and functional inputs. Three different GP models: (GP with scalar inputs, GP both scalar and functional inputs and GP with functional inputs) were employed and trained on the WEPP simulated data. Weight and nugget parameters in the covariance matrix of the GP model were estimated by the maximum likelihood method i.e eBayes approach. Particularly, we assumed weight parameters are built on trigonometric basis vectors.

GP model with functional inputs showed the best performance, while it's is the most computationally expensive among three approaches. This model predicted almost the same $\tilde{y}$ values as "true" y values for the inputs from the WEPP simulated training dataset.

CHAPTER 1. Introduction

Computer models are the implementation of complex mathematical models used to study many areas of scientific research (Sacks et al., 1989). It is a process of running large computer codes with various inputs to learn the output of dynamic real systems. Since often computer models solve systems of differential equations, they produce outputs that are functions of time as well as they require time-indexed inputs (Morris, 2014). Deterministic computer models $y = f(x)$, where x is finite-dimensional vector or scalar input, and y is also a finite dimensional vector or scalar output, are often computationally complex. Here the remedy is to build a "meta-model" or "surrogate" using less number of inputs. It can be used to quickly predict what the outputs for the new input values will be. Taking into account the scalar and vector nature of inputs and outputs, building such surrogate can be achieved using Gaussian processes (GPs).

1.1 Gaussian processes

A Gaussian process (GP) is a collection of time- or space- dependent random variables. Any finite collection of such random variables gives the multivariate normal distribution. Hence, GP is a generalization of the Gaussian probability distribution. Whereas a probability distribution describes random variables which are scalars or vectors (for multivariate distributions), a stochastic process governs the properties of functions (Williams and Rasmussen, 2006).

GPs provide an alternative approach to regression and classification problems. Given the same structure $y = f(x)$, we try to find the distribution over all functions $f(x)$, where those functions are consistent with the observed data. Here comes the Bayesian approach, as we set prior distribution for functions, update it with the observed data and get the posterior distribution over functions.

Choosing prior distribution for functions depends on the constraints and characteristics of the data we observed, as well as on the prior knowledge on a specific dynamic real system we have. For simplicity, most of the time we set the prior mean of functions to be zero. Other aspects of functions like stationarity and smoothness are controlled by various parameters, which are contained in the covariance function (kernel). We use specific covariance function $k(x, x')$ to ensure that the input values which are close to each other will at some point output values which are also close, respectively.

Covariance functions are built on various kernel types, which are controlled by different tuning parameters. The choice of the kernel is based on the nature of observed data and the research problem. As well as the choice of distance metric (eg. Euclidean) also varies by the problem.

This is what makes it appealing in the Gaussian processes. They often have characteristics that can be changed by setting certain parameters. Due to this flexible nature of GPs, we don't have to worry if it is possible to fit a model to certain data. It makes GPs to be non-parametric and ideal to fit almost any non-linear models.

1.1.1 Gaussian processes with functional inputs

In this research, we are interested in the computer models which built on the systems of ordinary or partial differential equations and require time- or space- indexed inputs. GPs by their structure are one of the suitable models for such time- or space- varying functional input systems. These type of computer models often force the output to be functional too. The functional outputs can be reduced to a lower dimension for further analysis using some dimension reduction techniques (Morris, 2014).

CHAPTER 2. Data

Data was collected by running the Water Erosion Prediction Project (WEPP) computer model. We used various input files {*climate, management, slope, soil*}, which are required for running the computer model. All input files were collected according to WEPP documentation standards. We also took into account the location of Science-based Trials of Rowcrops Integrated with Prairie Strips-1 (STRIPS-1) Project sites, s.t. WEPP inputs are specific for the sites of this project only. STRIPS-1 Project is composed of a team of scientists, educators, farmers, and extension specialists working on the prairie strips farmland conservation practice (Zhou et al., 2010)

2.1 WEPP computer model

WEPP model, is a soil erosion and runoff prediction technology used by the United States Department of Agriculture (USDA). WEPP was primarily used by USDA-Soil Conservation Service, USDA-Forest Service and USDI-Bureau of Land Management as it was initially released in 1995 (Flanagan et al., 2007). It is an improved prediction technology based on modern hydrologic and erosion science, process-oriented, and computer-implemented. WEPP model is a continuous simulation computer program that predicts erosion by considering various functional, quantitative and categorical inputs (e.g., precipitation amounts and its intensity, soil textural qualities, plant growth parameters, residue decomposition parameters, effects of tillage implements on soil properties and residue amounts, slope shape, steepness, and orientation, and soil erodibility parameters etc.). Continuous simulation means WEPP simulates consecutive years (maximum is 12) and each year has specific input daily climate data. Also, the computer model can be applied for both complex watershed models or a single hillslope model.

2.2 WEPP simulation analysis

For the simplicity and consistency of parameters set, we run WEPP simulation analysis separately for a single hillslope, and in total for 36 hillslopes. STRIPS-1 Project was conducted on a Neal Smith National Wildlife Refuge (NSNWR) site, which contains 12 small agricultural watersheds {*Basswood1, Basswood2, Basswood3, Basswood4, Basswood5, Basswood6, Interim1, Interim2, Interim3, Orbweaver1, Orbweaver2, Orbweaver3*}. Each watershed consists of 3 hillslopes, with the exception of *Interim2* and *Interim3*, which consist of 4 and 2 hillslopes, respectively. Division of watersheds was based on the homogeneity of the soil texture, hence we used specific soil input file for every single hillslope. Altogether single WEPP model run requires a climate input file, which is constant throughout 36 hillslopes, management input file, which is also constant for all hillslopes, slope file, which is hillslope specific, and soil texture input file, which is also individual for each hillslope.

There is a probability for runoff for each simulation day. If storm occurred and WEPP predicts a runoff event, it is recorded to one of the output files. WEPP produces many different kinds of outputs, which include detailed information on soil, plant, water balance, crop yield, rangeland, etc. The output file of our interest **.env* contains average values of soil deposition, detachment, sediment delivery, and enrichment. Annual soil loss (detachment) distributions in various locations are given in Figures 2.1 and 2.2. In Figure 2.2 we can see some periodicity in the distribution of soil loss with a peak amount of soil loss more than 0.3 kg/m^2 in 2010 and 2015. Although in Figure 2.1 there is a lot of similar soil loss amount presumably due to similar precipitation, there are some differences as well, which suggests that there might be interactions between input variables. For instance, Orbweaver2 hill3 has similar soil loss in 2012 and 2013, while Basswood1 hill1 has more soil loss in 2013 than in 2012.

2.2.1 Climate

The Iowa Environmental Mesonet curates many datasets attempting to provide environmental data for Iowa and beyond. Lots of scripts and workflows run each day to churn out

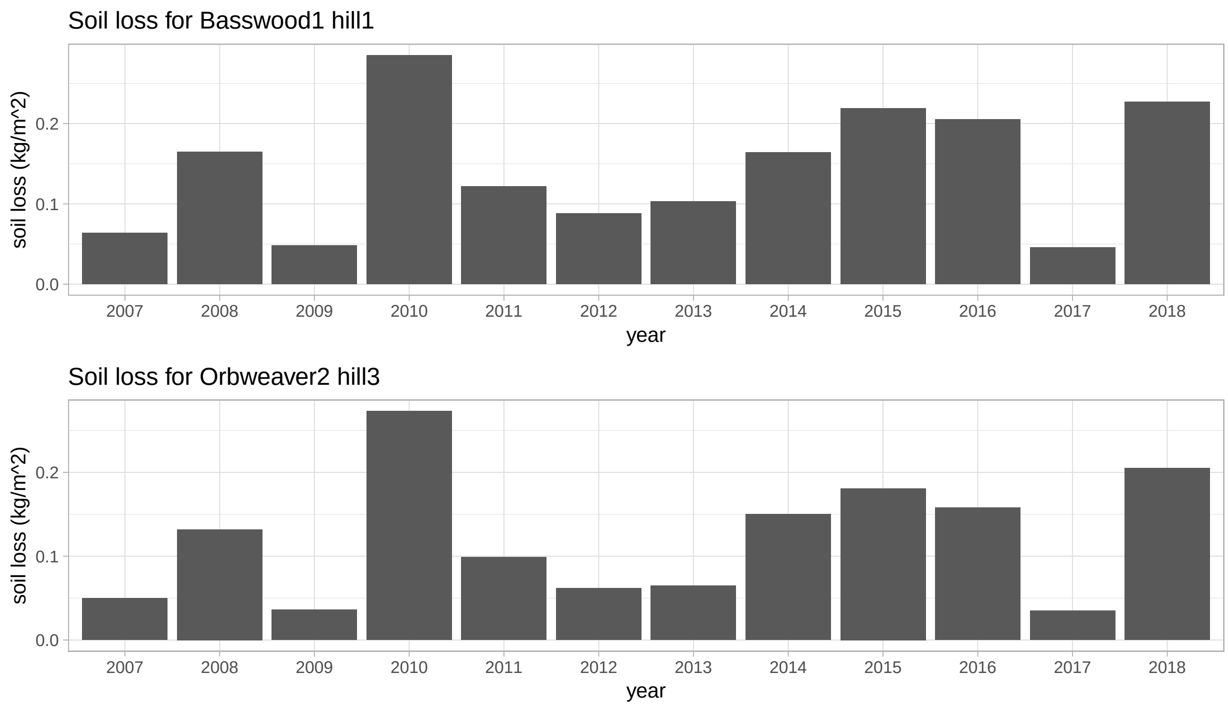

Figure 2.1 WEPP simulated annual soil loss amount from 2007 until 2018 for hillslopes in agricultural watersheds Basswood1 and Orbweaver2.

products, one of which are the daily climate files suitable for WEPP. Climate data was retrieved from the nearest station close to NSNWR. Alternatively, climate file can be easily generated by CLIGEN program, a stochastic weather generator which produces daily estimates of precipitation, temperature, dewpoint, wind, and solar radiation for a single geographic point, using monthly parameters (means, SD's, skewness, etc.) derived from the historic measurements. CLIGEN requires station parameter files for a run, and a user can choose from over than 1000 climate stations across the US.

The data file contains daily records on daily precipitation (mm), duration of precipitation (h), daily maximum and minimum temperature (C), dew point temperature (C), ratio of time to rainfall peak/rainfall duration, ratio of maximum rainfall intensity/average rainfall intensity, wind direction (degrees from North), wind velocity (m/sec), daily solar radiation (langleys/day). WEPP allows maximum 12 years of simulation, thus the data file contains daily climate records starting from 2007 to 2018 only.

For simplicity, we started with a small and fixed number of inputs. Hence we only used

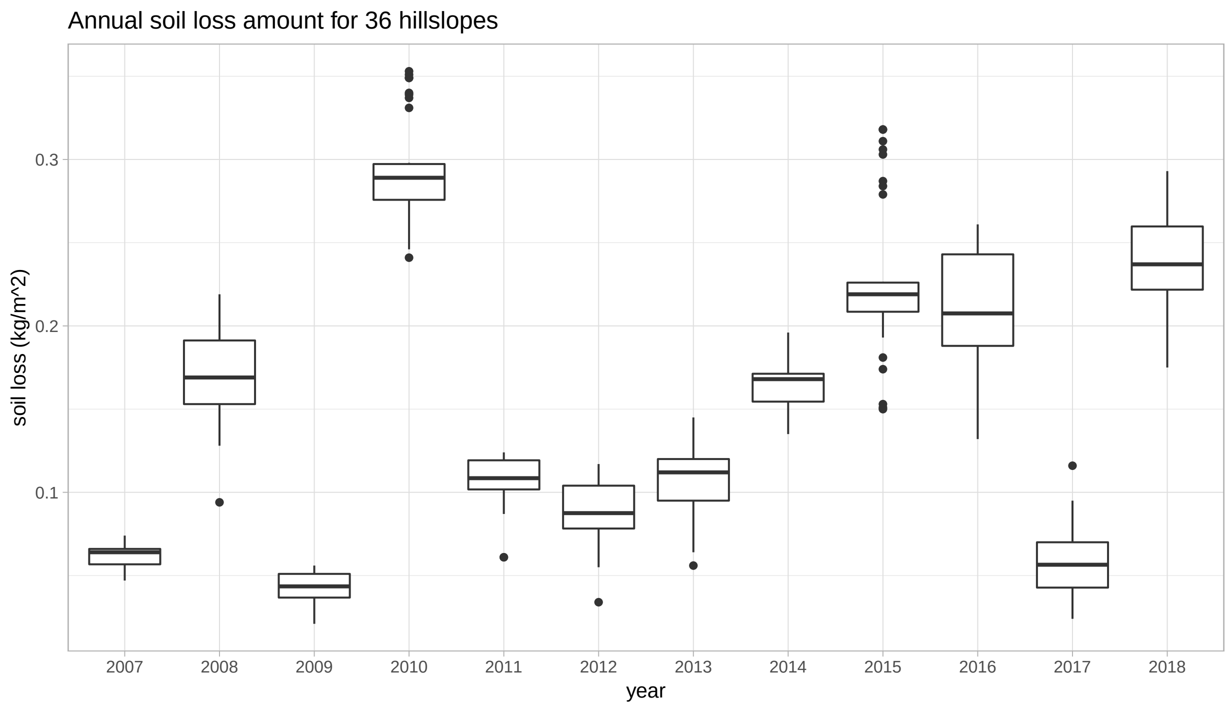

Figure 2.2 WEPP simulated annual soil loss amount from 2007 until 2018 for all 36 hillslopes in STRIPS-1 Project agricultural watersheds in NSNWR.

precipitation from climate file as the main input. As the next steps of our research, we can add more variables from the climate file. The plot in Figure 2.3 shows the distributions of daily precipitation for 2007 and 2018 year. It is depicted from the data collected from a climate station close to NSNWR. In Figure 2.3 we see that two years have different precipitation profiles. In 2007 there are multiple days with the precipitation amount more than 40 mm in different seasons, while in 2018 there is a peak of precipitation with more than 40 mm only in summer. We also added precipitation distributions for all years from 2007 to 2018 in Figures A.1 and A.2 (Appendix 1).

2.2.2 Management

The management input file contains all of the information needed for the WEPP simulation analysis related to plant parameters (rangeland plant communities and cropland annual and perennial crops), tillage sequences and tillage implement parameters, plant and residue

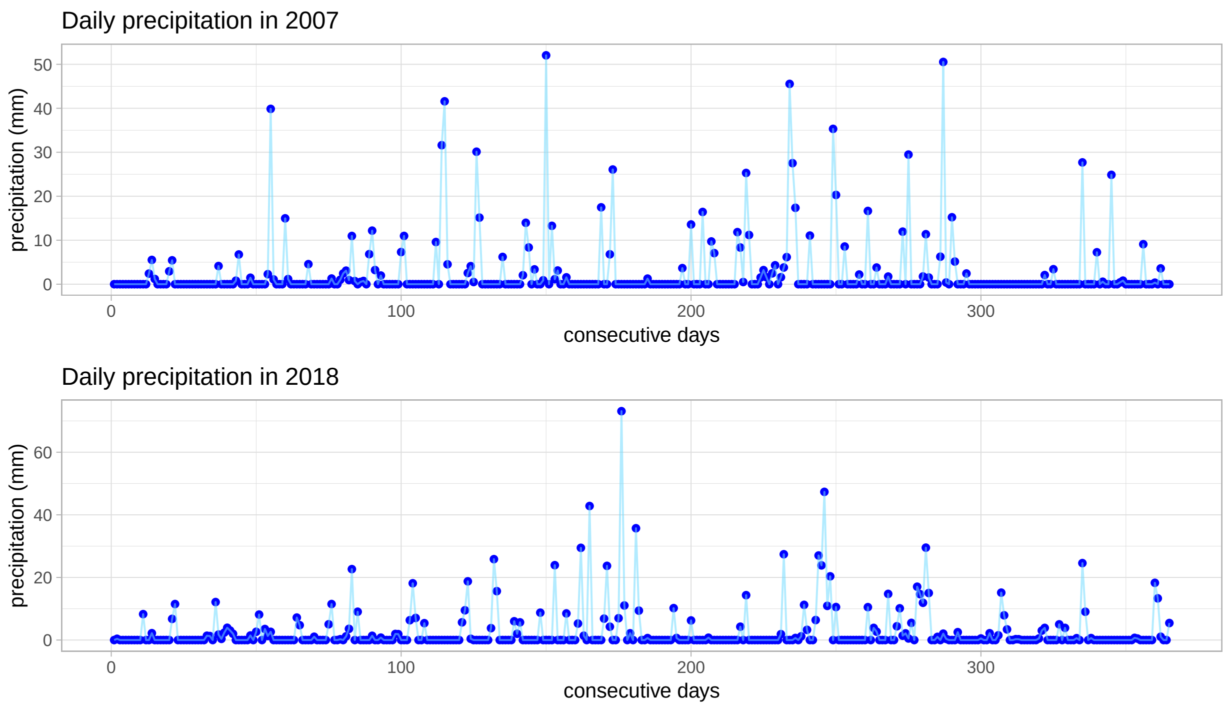

Figure 2.3 Daily precipitation amount for years 2007 and 2018. Data is collected from the NSNWR neighborhood climate station.

management, initial conditions, contouring, subsurface drainage, and crop rotations.

We used the same "corn-soybean-no-till" file for all 36 watershed hillslope simulations. Crop rotation technique used in Iowa can help to reduce soil erosion on sloping lands and improve soil health by adding diverse biological activity (STYLE, 2005). For the crop rotation we chose *corn* and *soybean*. Corn yield is improved if it's rotated with some other crop. In addition, corn and soybean are used because they are the two most profitable crops. For the initial conditions, we set the rotation cycle to start with *soybean* from the 2007 year. If the timing details are available, s.t. tilling date, fertilizing date, pesticide application date, and harvest date, one can add them to the management file. We used *no tillage* and kept all other parameters the same as WEPP's default parameter values.

2.2.3 Slope

The slope file contains information about the landscape geometry, like slope orientation, slope length, and slope steepness at points down the profile. For the WEPP simulation analysis, we used 36 hillslope specific slope files. Hillslope lengths range from 5.98 to 125.60 meters. The slope is measured for every 5 meters. For the consistency purposes we divided the hillslope length into $n = 15$ equal sized sublengths and recorded corresponding slope steepness as a numerical input at n_i^{th} sublength for $i \in \{1, 2, 3, ..., 15\}$. We used "face" transect of the hillslope. Another "gulley" transect can be used later for sensitivity analysis of emulator on the transect type.

Slope profiles are depicted from two different perspectives and given in the following Figures 2.4 and 2.5. Basswood1 hill1 in Figure 2.4 has a more gentle slope profile than Orbweaver2 hill3 depicted in Figure 2.5. For a better visual analysis, we also attached the slope profiles of each watershed in Figures A3 to A14 (Appendix 1). Note that although slope value is positive, the direction is negative.

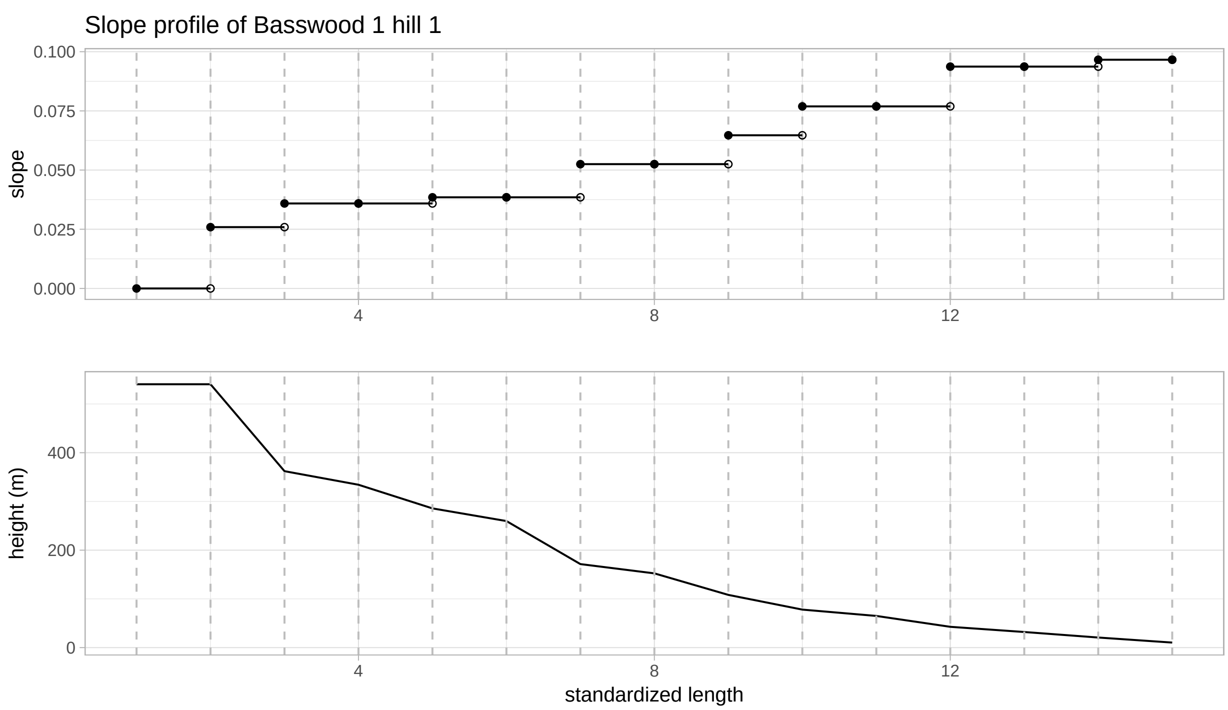

Figure 2.4 Slope profile of the hillslope 1 in Basswood1 watershed.

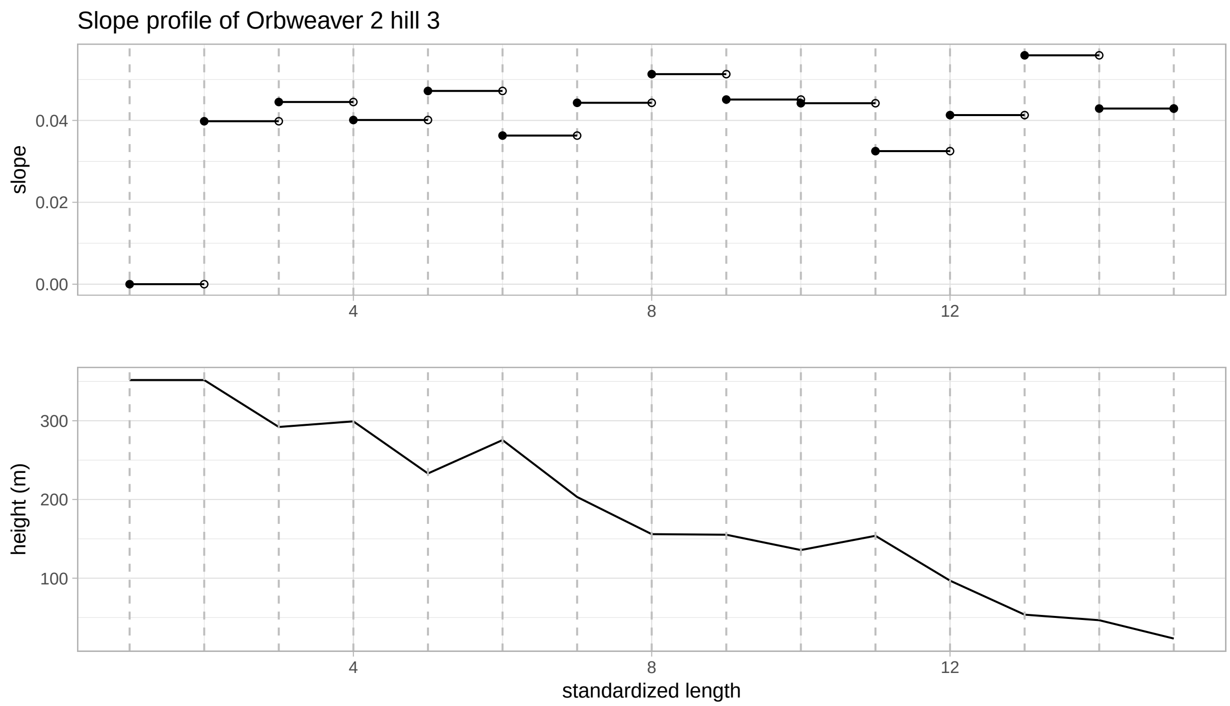

Figure 2.5 Slope profile of hillslope 3 in Orbweaver2 watershed.

2.2.4 Soil

There are individual hillslope specific soil input files used for the WEPP simulation analysis. Data files are taken from the SSURGO database provided by USDA NRCS and in turn, contain digital soil maps and accompanying detailed soil properties. The database is collected by the National Cooperative Soil Survey over the course of a century. Soil information was gathered simply by walking over the land and observing the soil. Soil samples were collected and later analyzed in laboratories for their structure (NRCS, 2010). An example of the detailed soil structure is given in Table 2.1. It contains information on depth from soil surface to bottom of soil layer (mm), percentage of sand in the layer, percentage of clay in the layer, percentage of organic matter (volume) in the layer, cation exchange capacity in the layer (meq/100 g of soil), percentage of rock fragments by volume in the layer.

Soil input files differ by state/county. For instance, *LADOGA.SIL* is a data file designed for *Jasper* county in *Iowa* state. Note that all 12 watersheds in STRIPS-1 Project are located in NSNWR, which in turn is located in Jasper county of Iowa. There are in total 8 soil texture types present in STRIPS-1 watersheds. Thus there are 8 soil input files:

Depth(mm)	Sand(%)	Clay(%)	Organic(%)	CEC(meq/100g)	Rock(%)
254.00	6.40	24.50	2.50	24.00	0.00
1143.00	22.00	39.00	0.83	31.20	0.00
1524.00	25.70	28.00	0.28	22.40	0.00

Table 2.1 Detailed LADOGA(SIL) soil texture information measured to a maximum depth of 1.8 meters.

ACKMORE(SIL), ARMSTRONG(L), CLARINDA(SICL), GARA(L), LADOGA(SIL), LAMONI(L), OTLEY(SICL), SHELBY(L). Those soil types were allocated in watersheds by latitude and longitude coordinates provided in data files. Note that watersheds were divided into hills of various lengths according to the soil type. For simplicity, we kept *soil* input constant for each individual hillslope and didn't include this information to the emulator. As a next step, we can convert soil texture information given in Table 2.1 to the various functional numerical inputs which can be used as additional input variables in GP emulator.

2.3 Scientific questions related to WEPP

Winds and rainstorm are the predominant environmental conditions that account for the major soil loss in Iowa. It carries off tons of rich, fertile topsoil from the farmland and results in potentially $1 billion cuts in yield in state's 88,000 farms (Eller, 2014).

Organizations use various tools to control and reduce soil loss in Iowa. There are many models for quantifying soil loss. WEPP computer model, in turn, is widely used in academia and industry. Although WEPP gives the most precise estimate of soil loss, it requires massive input data.

Our ultimate goal in this research is to build a computer model-agnostic GP emulator. For instance, we want to build an emulator for the WEPP computer model using only the major input variables it requires for simulation.

CHAPTER 3. Methods

3.1 Gaussian processes

As was mentioned earlier, the Gaussian process is a collection of random variables, any finite number of which have a Gaussian distribution. GP is completely defined by its mean and covariance structure (Williams and Rasmussen, 2006).

Consider a simple GP with input x and output y, where y is a collection of random variables s.t. $y = \{y_1, y_2, ..., y_n\}$ for the corresponding collection of input values $x = \{x_1, x_2, ..., x_n\}$. There is some *true* function f, for which $f(x_i) = y_i$ for some $i \leq n$. Often, input x_i is time indexed and $x_i \in \mathbb{R}^D$. As an alternative to the linear regression models, we can use GPs to make inference about f directly from the functional space. Let $f \sim GP_x(\mu, k)$ denote that f is a GP with input space x s.t. for some $i, j \leq n$, $E[f(x_i)] = \mu(x_i)$ where $\mu(x)$ a mean function and $Cov(f(x_i), f(x_j)) = k(x_i, x_j)$ where $k(x, x')$ is a covariance function. Using the covariance function, we build a covariance matrix:

$$\Sigma = \begin{bmatrix} Cov(f(x_1), f(x_1)) & Cov(f(x_1), f(x_2)) & ... & Cov(f(x_1), f(x_n)) \\ Cov(f(x_2), f(x_2)) & Cov(f(x_2), f(x_2)) & ... & Cov(f(x_2), f(x_n)) \\ ... & ... & & \\ Cov(f(x_n), f(x_1)) & Cov(f(x_n), f(x_2)) & ... & Cov(f(x_n), f(x_n)) \end{bmatrix}$$

Covariance function is a crucial ingredient in GPs, as it encodes our assumptions about the function we would like to learn. The specification of the covariance function implies a distribution over functions (Williams and Rasmussen, 2006). Since we want the input values that are near in input space map to the similar output value, our covariance function should have *nearness* property. Covariance functions have non-negative definiteness property as well, which makes the Σ a valid covariance matrix. There are four aspects of the Gaussian processes which can be defined through the covariance functions: $\{stationarity, isotropy, smoothness, periodicity\}$.

Stationary covariance function is the function of $x - x'$ and it is invariant to any translations in the sample input space. The most common stationary covariance function is

$$Squared - Exponential : k(x, x') = exp\big(-\frac{(x - x')^2}{2l^2}\big)$$

Here parameter l is the characteristic length-scale of the Gaussian process.

Isotropic covariance function is the function of $|x - x'|$ and it is invariant to all rigid motions in sample input space. The most common isotropic covariance functions are

$$Squared - Exponential : k(x, x') = exp\big(-\frac{(x - x')^2}{2l^2}\big)$$

$$Matern : k(x, x') = \frac{2^{1-v}}{\Gamma(v)}\big(\frac{\sqrt{2v}|x - x'|}{l}\big)^v K_v\big(\frac{\sqrt{2v}|x - x'|}{l}\big)$$

$$Ornstein - Uhlenbeck : k(x, x') = exp\big(-\frac{|x - x'|}{l}\big)$$

Here K_v is the modified Bessel function of order v and $\Gamma(v)$ is the Gamma function evaluated at v. And l is characteristic length-scale of the process.

There is a wide range of choice for covariance functions, which are periodic, non-stationary, anisotropic, linear, polynomial, piecewise-polynomial and etc.

Non-negative definiteness property of covariance functions implies a valid covariance matrix Σ. However, in an application, if the input values in the sample dataset happen to be similar (too close to each other in input space $\mathbb{R}^D$) it leads to the singularity of the covariance matrix. The common remedy would be *regularization* of that matrix. Adding a penalty term also known as *nugget* to the principal diagonal: $\Sigma + \lambda I$ will stabilize the matrix. This idea is borrowed from the concept of *Ridge* regression models.

By its definition, GPs follow the marginalization property (Williams and Rasmussen, 2006). For instance, consider a collection of random variables y. GP specifies

$$y \sim N_n(\mu, \Sigma)$$

where $\mu = (\mu_1, \mu_2, ..., \mu_n)$ and $\Sigma_{n \times n}$ is a covariance matrix. The marginal distributions are given as $y_i \sim N(\mu_i, \Sigma_{ii})$ and $y_j \sim N(\mu_j, \Sigma_{jj})$. Likewise we can easily obtain condional distribution of y_i on y_j.

For the prediction of new output values in GPs we can employ the same technique and derive posterior joint distribution of *f(x)* $= y$ from the prior distribution and observed data:

$$\begin{pmatrix} y \\ \tilde{y} \end{pmatrix} \sim N\left(0, \begin{pmatrix} \Sigma & \widetilde{\Sigma} \\ \widetilde{\Sigma}^T & \widetilde{\widetilde{\Sigma}} \end{pmatrix}\right)$$

For simplicity, we let the prior mean $\mu(x) = 0$. Here y is the observed data and $\tilde{y}$ is new output values for the corresponding new input values $\tilde{x}$. Hence the conditional distribution of $\tilde{y}$ on the prior distribution and observed data is given as the follows:

$$\tilde{y}|\tilde{x}, x, y \sim N(\widetilde{\Sigma}^T\Sigma^{-1}y, \widetilde{\widetilde{\Sigma}} - \widetilde{\Sigma}^T\Sigma^{-1}\widetilde{\Sigma})$$

To be able to do prediction we should estimate parameters in the covariance matrix Σ. This can be done in multiple ways, using MCMC simulations or employing an empirical Bayes (eBayes) approach. In the latter approach, hyperparameters (parameters involved in the covariance structure of prior distribution) are estimated using the observed data by maximum likelihood estimation (MLE) method.

3.2 Gaussian processes with functional inputs

Using the generalization of GP "meta-models" with functional inputs defined by (Morris, 2012), we have the following design. *Correlation distance* for inputs $x_i, x_j \in \mathbb{R}^T$ through some temporal or spatial indices $[1, T]$ is defined as $d(x_i, x_j) = \omega^\top |x_i - x_j|^\alpha$ for $\alpha \in (0, 2]$. Here the weight parameter ω is defined over $[1, T]$. Later we chose $\alpha = 2$, which leads to the most popular covariance function, which is in the case is appropriate for models that implement systems of partial differential equations (Morris, 2014). We assume that $\omega_t \in \mathbb{R}^+$ and has some upper bound. From analyzing the WEPP model, we came into conclusion that weight parameter has wiggly periodic structure. The same assumption also serves as a remedy for issues like "curse of dimensionality" that might arise in further model fitting.

In this paper we consider different models which have both scalar and functional inputs. For a finite collection of output random variables $y = \{y_1, y_2, ..., y_n\}$ we have corresponding input variables $x = \{x_1, x_2, ..., x_n\}$, where $x_i \in \mathbb{R}^D$. Particularly, for $i \leq n$ where $x_i =$

$(x_{i,1}, x_{i,2}, ..., x_{i,D^*})$ s.t. $x_{i,d} \in \mathbb{R}^{T_d}$ and $\sum_{d=1}^{D^*} T_d = D$. Part of the inputs are assumed scalar if $T_d = 1$ and some are assumed functional if $T_d > 1$. Note that if functional input has dimension T_d then its corresponding weight parameter $\omega_d \in \mathbb{R}^{T_d}$ where $\omega = (\omega_1, \omega_2, ..., \omega_{D^*})$. When $D >> n$ i.e parameter dimension is too high, we fall into "curse of dimensionality". We propose a possible remedy which is described in details later.

For the GP emulator, we chose a $squared - exponential$ covariance function, which is stationary and isotropic. For some $i, j \leq n$ s.t. $i \neq j$

$$k(x_i, x_j) = e^{-\omega^\top (x_i - x_j)^2}$$

and when $i = j$

$$k(x_i, x_j) = 1 + \lambda$$

Here ω is the weight parameter, which is proportional to the inverse of length-scale parameter l. It also has the same dimension as x_i, s.t. $\omega \in \mathbb{R}^D$. Based on the complexity of input x_i, we have different parameters and corresponding parameters spaces.

3.2.1 Weight parameters built on trigonometric basis functions

We would like to add some penalty to the model by structuring the weight parameters separately. This would serve as a possible remedy for "curse of dimensionality". Under the assumption that x is a functional input, we will force a weight function (Morris, 2014) used in dynamic linear models.

Consider a *Fourier frequencies*

$$w_q = \frac{2\pi q}{T}, \quad q = 0, 1, ..., \frac{T}{2}$$

and also the following T-dimensional vectors:

$$e_0 = (1, 1, ..., 1)'$$

$$c_1 = (\cos w_1, \cos 2w_1, ..., \cos T w_1)'$$

$$s_1 = (\sin w_1, \sin 2w_1, ..., \sin T w_1)'$$

$$...$$

$$c_q = (\cos w_q, \cos 2w_q, ..., \cos Tw_q)'$$

$$s_q = (\sin w_q, \sin 2w_q, ..., \sin Tw_q)'$$

$$...$$

$$c_{T/2} = (\cos w_{T/2}, \cos 2w_{T/2}, ..., \cos Tw_{T/2})'$$

where q is the number of harmonics. We should note that the last vector $s_{T/2}$ is not written since it is vector of zeroes. There are in total T vectors of length T. One can also show that these vectors are orthogonal, which in turn implies that every vector $\omega \in \mathbb{R}^T$ can be written as a linear combination of $e_0, c_1, s_1, ..., c_{T/2}$:

$$\omega = \alpha_0 e_0 + \sum_{q=1}^{T/2-1} (\alpha_q c_q + \beta_q s_q) + \alpha_{T/2} c_{T/2}$$

Thus these vectors form a basis for a vector $w \in \mathbb{R}^T$. One can also choose the number of harmonics q, thus restrict the basis for a vector ω.

This approach is widely described in (Petris et al., 2009). We assume it is effective when $T >> n$ or even when $T >> 10$. Number of parameters which have to be estimated decrease gradually from T to $2q + 1$, where q is the number of harmonics used in creating a basis for ω. Hence new parameters of interest are the coefficients of basis vectors:

$$\theta = (\alpha_0, \alpha_1, \beta_1, \alpha_2, \beta_2, ..., \alpha_q, \beta_q)$$

Finally, we have three proposed GP models for building an emulator for the WEPP computer model. Models share the same prior mean $\mu(x) = 0$ and differ by the x input variable structure and corresponding parameters in covariance function $k(x, x') = e^{-\omega^\top (x-x')^2}$:

1. For $x_i \in \mathbb{R}^D$, where $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,D^*})$ s.t. $x_{i,d}$ is scalar $\forall d \in \{1, 2, ..., D^*\}$, and $w = (w_1, w_2, ..., w_D^*)$ s.t. $w_d \in \mathbb{R}$. Hence $T_d = 1$ $\forall d \in \{1, 2, ..., D^*\}$ and $D = D^*$.

2. For $x_i \in \mathbb{R}^D$, where $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,D^*})$ and $x_{i,d} \in \mathbb{R}^{T_d}$. For some finite collection $F = \{d : T_d = 1\}$ $x_{i,d}$ is scalar and for the other finite collection F^c, which is complement of F, all $x_{i,d}$ are functional s.t. the dimension T_d is greater than one. Corresponding $\omega = (\omega_1, \omega_2, ..., \omega_{D^*})$ has the same structure and dimension. For any $d \in F$, $\omega_d \in \mathbb{R}$ and for any $d \in F^c$, $\omega_d \in \mathbb{R}^{T_d}$ s.t. $T_d > 1$. Note that $\sum_{d=1}^{D^*} T_d = D$ and $F \cup F^c = \{1, 2, ..., D^*\}$.

3. For $x_i \in \mathbb{R}^D$, where $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,D^*})$, $x_{i,d}$ is functional $\forall d \in \{1, 2, ..., D^*\}$ s.t. $T_d > 1\ \forall d$. Corresponding $w = (w_1, w_2, ..., w_D^*)$ has the same structure with $w_d \in \mathbb{R}^{T_d}$ s.t. $T_d > 1$. Likewise, $\sum_{d=1}^{D^*} T_d = D$.

CHAPTER 4. Analysis

We ran three different Gaussian process models to build an emulator for the WEPP computer model. We used the data generated from the WEPP simulation analysis as a primary dataset. For each case, input variables were modified accordingly. MLE estimates of the target parameters are given for each model. Also, we attached various plots depicting original data and the data generated by GP emulator for a better visual analysis.

4.1 Analysis on simulated data

4.1.1 Simulating data

Firstly, we ran a GP model on a simulated data. There are some assumptions that we made about the input variable x. Using the Gaussian processes concept, we simulated data in the following manner:

1. Simulate $x_i \sim N_T(0, \Omega)$ for $i = 1, ..., n$, where n is the sample size and x_i is a vector of length T, i.e $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,T})$. Here Ω has first order autoregressive structure AR(1) with unit variance $\sigma = 1$ and $\rho = 0.99$ s.t.

$$\Omega = \begin{bmatrix} 1 & \rho & \rho^2 & ... & \rho^{n-1} \\ \rho & 1 & \rho & ... & \rho^{n-2} \\ \rho^2 & \rho & 1 & ... & \rho^{n-3} \\ & ... & & ... & \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & ... & 1 \end{bmatrix}$$

2. Next, we simulate weight parameter ω for x s.t. $\omega \in \mathbb{R}^T$ and ω has a periodic structure. For convenience we set $q = 1$ and generate basis e_0, c_1, s_1 for ω. Coefficients $\alpha_0, \alpha_1, \beta_1$ for the basis vectors are randomly generated from *Uniform(-30,30)* distribution.

3. We also randomly generate *nugget* parameter λ from *Uniform(0, 0.1)* distribution.

4. As a next step we calculate a $n \times n$ covariance matrix Σ. Using the covariance function given in *Methods* chapter, s.t. for any $i, j \leq n$ and $i \neq j$, $Cov(f(x_i), f(x_j)) = k(x_i, x_j) = e^{\omega^\top (x_i - x_j)^2}$ and for $i = j$: $k(x_i, x_j) = 1 + \lambda$, we fill each cell in Σ.

5. Finally, the response variables $y = (y_1, y_2, ..., y_n)$, where y_i is assumed to be scalar, are simulated from another Gaussian Process model s.t.

$$y \sim N_n(0, \Sigma)$$

4.1.2 Gaussian Process model with functional input

For simplicity, we test the GP model with one functional input and the scalar output, where $y = \{y_1, y_2, ..., y_{300}\}$ s.t $y_i \in \mathbb{R}$ and $x = \{x_1, x_2, ..., x_{300}\}$ s.t. $x_i \in \mathbb{R}^{20}$ and $f(x_i) = y_i$. Corresponding weight parameters $\omega = (\omega_1, \omega_2, ..., \omega_{20})$ have periodic structure. Thus the target parameters are the coefficients $\theta = (\alpha_0, \alpha_1, \beta_1, ..., \alpha_q, \beta_q)$ of the basis vectors which are used in the construction of weight parameters. We also added *nugget* parameter to the diagonal of the covariance matrix, set prior mean $\mu(x) = 0$ and set $q = 1$, to have:

$$\theta = (\alpha_0, \alpha_1, \beta_1, \lambda)$$

Applying eBayes approach, we found the MLE estimate of theta:

$$\hat{\theta}_{MLE} = (-29.75041863, -2.21265674, 2.60807170, 0.03704142)$$

We plugged in estimated parameters and got predicted values $\tilde{y}$ for a grid of first 50 simulated input values $\tilde{x}$. Distribution of the corresponding response variable values y and $\tilde{y}$ for $x_1, x_2, ..., x_{50}$ are plotted in Figure 4.1.

4.2 Analysis on the real WEPP data

4.2.1 Gaussian Process model with scalar inputs

In the first Gaussian process model we treat both inputs as scalar as well as the output random variable. We set $y = \{y_1, y_2, ..., y_{432}\}$ where y_i is annual total soil loss amount in a

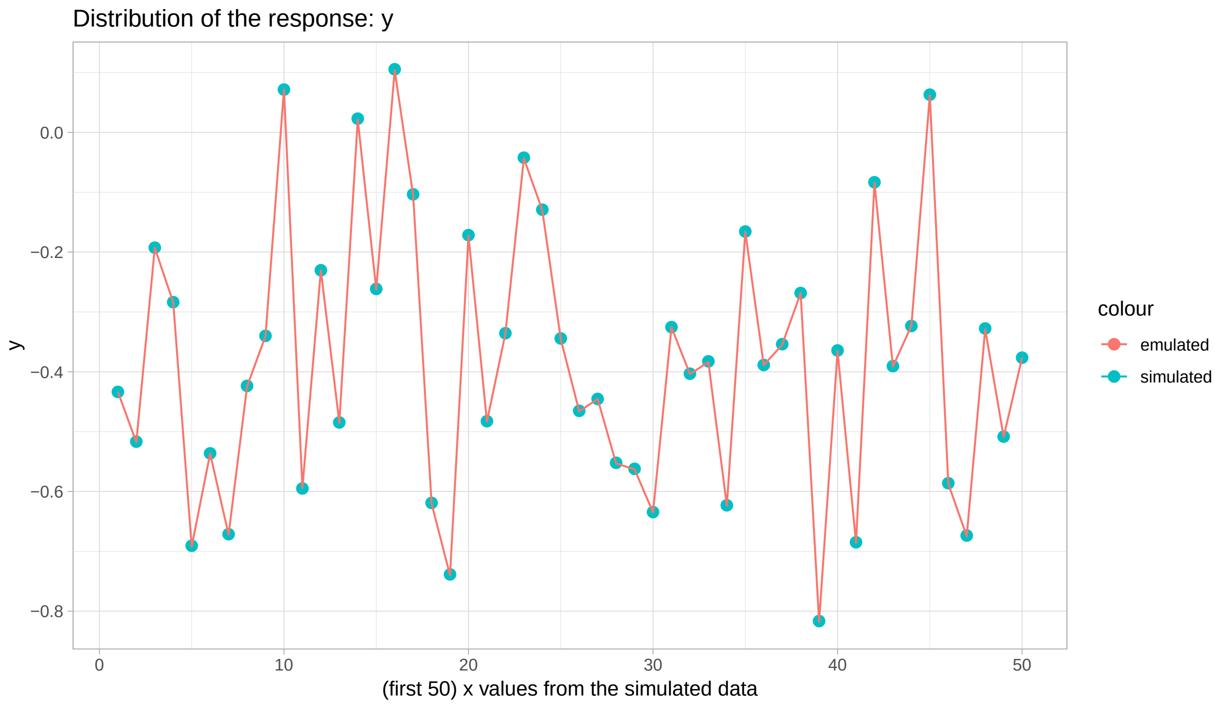

Figure 4.1 Distribution of the simulated and emulated response variable y for the first 50 simulated x values.

specific hillslope. Since there are 12 years from 2007 until 2018 and 36 different hillslopes among 12 agriculutural watersheds, we have in total $12 * 36 = 432$ observations. Hence the sample size is $n = 432$. Input variable $x = \{x_1, x_2, ..., x_{432}\}$ has two components. For some $i \leq n$, $x_i = (x_{i,1}, x_{i,2})$, where $x_{i,1} \in \mathbb{R}$ is the annual total precipitation amount in the Neal Smeath region and $x_{i,2} \in \mathbb{R}$ is the average slope of a specific hillslope.

As for the most of the cases, we let prior mean to be zero $\mu(x) = 0$. The range of precipitation amount is very big which results in a multiple zero entries in a covariance matrix. We have $e^{-d(x,x')} \approx 0$ for the most cases and it leads to degeneracy of the covariance matrix. As a remedy, we log-transformed the total precipitation amount. Taking into account the regularization *nugget* parameter, we have the following target parameter to be estimated:

$$\theta = (\omega_1, \omega_2, \lambda)$$

Applying eBayes approach, we found the MLE estimate of theta:

$$\hat{\theta}_{MLE} = (0.529489806, 0.123338935, 0.003394897)$$

For a grid of new input values $\tilde{x}$ and corresponding $\tilde{y}$, observed data y and MLE estimates, we have:

$$\tilde{y}|\tilde{x}, x, y \sim N(\widetilde{\Sigma}^T \Sigma^{-1} y, \widetilde{\widetilde{\Sigma}} - \widetilde{\Sigma}^T \Sigma^{-1} \widetilde{\Sigma})$$

Two heat maps of soil distribution from the data generated by WEPP and GP emulator are given in the following Figures: 4.2 and 4.3.

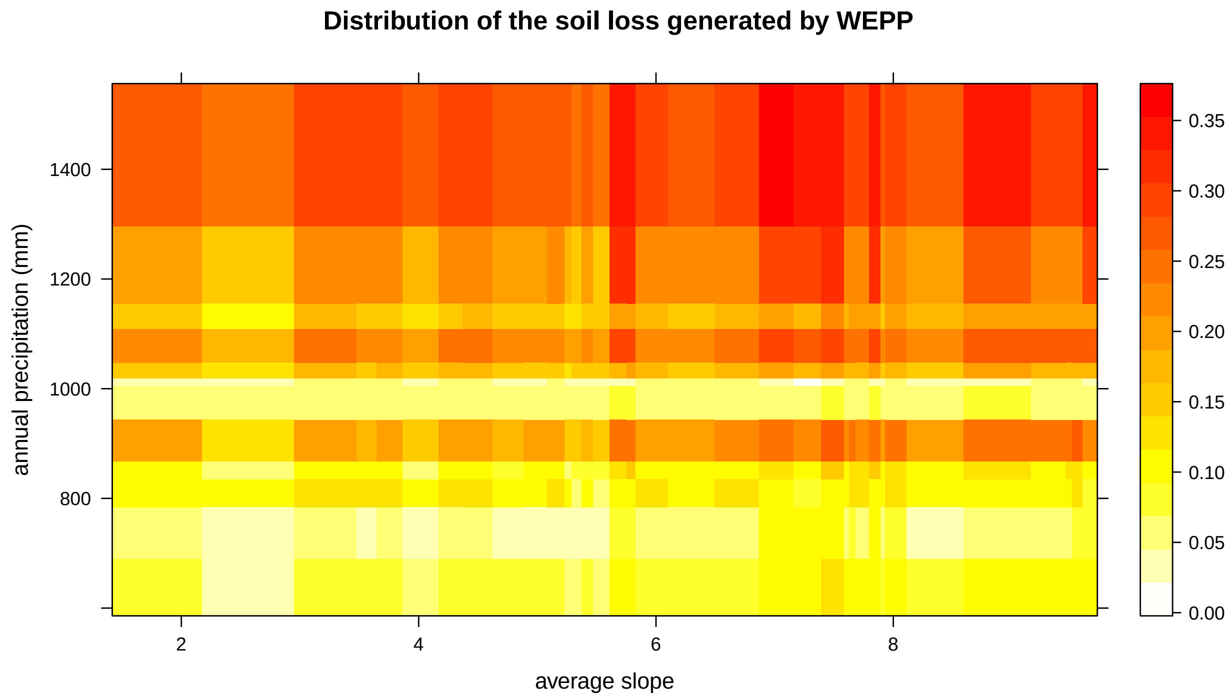

Figure 4.2 Soil loss distribution with respect to the total annual precipitation and average slope of a hillslope. Data is taken from WEPP simulation analysis.

4.2.2 Gaussian Process model with scalar and functional inputs

In this Gaussian process model we treat one input as scalar and the other one as a functional. The response $y = \{y_1, y_2, ..., y_{432}\}$ is still kept as scalar. Input variable $x = \{x_1, x_2, ..., x_{432}\}$ has two components. For some $i \leq n$, $x_i = (x_{i,1}, x_{i,2})$, where $x_{i,1} \in \mathbb{R}$ is the annual total precipitation amount in the Neal Smith region. And $x_{i,2} \in \mathbb{R}^{15}$ is the profile of one specific hillslope. For instance, in $x_{i,2} = (x_{i,2,1}, x_{i,2,2}, ..., x_{i,2,15})$, for $1 \leq j \leq 15$ $x_{i,2,j}$ is a slope for

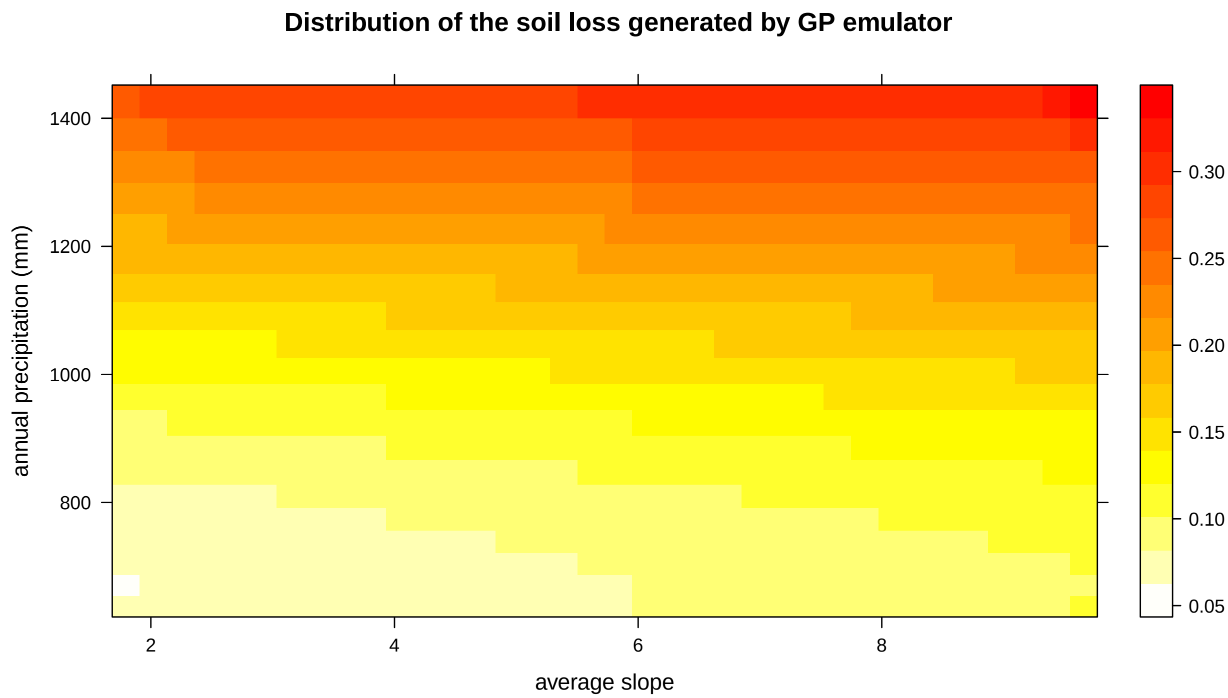

Figure 4.3 Soil loss distribution with respect to the total annual precipitation and average slope of a hillslope. Output values are generated by GP emulator designed by first approach model.

j^{th} partition of the length of a hillslope. Note that partitions are of the same length for each specific hillslope. Hillslope profiles are given in Figures A3 - A14 (Appendix 1).

We set the prior mean $\mu(x) = 0$. Number of harmonics in designing the weight parameters is chosen as $q = 1$. We log-transformed the precipitation amount and also added regularization *nugget* parameter to the diagonal of the covariance matrix. Thus the parameter

$$\theta = (\omega_1, \omega_2, \lambda)$$

where $\omega_2 \in \mathbb{R}^{15}$ is transformed to

$$\theta = (\omega_1, \alpha_0, \alpha_1, \beta_1, \lambda)$$

Applying eBayes approach, we found the MLE estimate of theta:

$$\hat{\theta}_{MLE} = (55.06167167, -402.44309188, -12.67886639, 386.78739175, 0.00197348)$$

Estimated target parameters were plugged into the GP model. Plots depicting soil loss distribution for the original WEPP data, and soil loss values predicted by the GP emulator for a grid of new annual precipitation amount values are given in Figures 4.4 and 4.5. 95% confidence intervals for the emulated soil loss values are also added. Since slope input is functional, we kept it constant while we vary the annual precipitation amount. Plots of the hillslope profiles used in the emulation part: *Basswood1 hill1 and Orbweaver2 hill3* are given in Figures 2.4 and 2.5 in *Data* chapter.

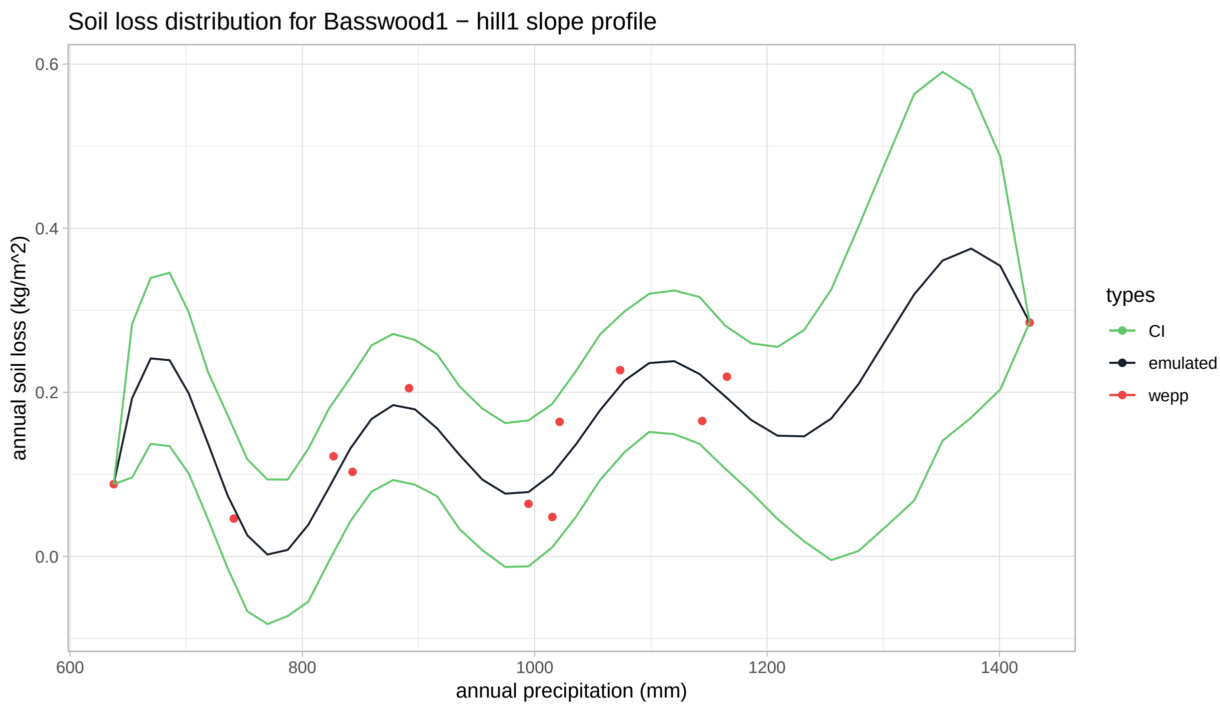

Figure 4.4 Soil loss distribution with respect to the total annual precipitation for Basswood1 hill1 slope profile. Data is taken from the WEPP simulation analysis.

4.2.3 Gaussian Process model with functional inputs

In the third Gaussian process model we treat both inputs as functional and the output random variable $y = \{y_1, y_2, ..., y_{432}\}$ as a scalar. Input variable $x = \{x_1, x_2, ..., x_{432}\}$ has two components. For some $i \leq n$, $x_i = (x_{i,1}, x_{i,2})$, where $x_{i,1} \in \mathbb{R}^{365}$ is a daily precipitation amount for one year and $x_{i,2}$ is the slope profile of one specific hillslope.

Here we also set the prior mean $\mu(x) = 0$. Due to the same reasons as in the first and second

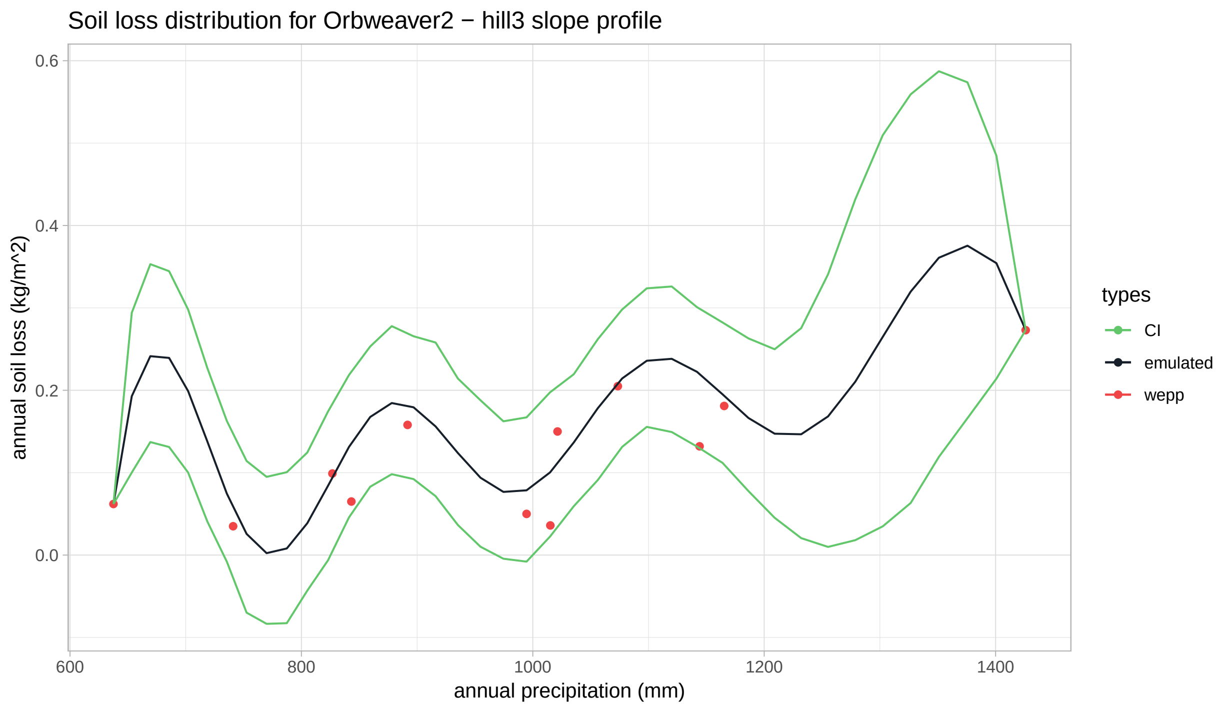

Figure 4.5 Soil loss distribution with respect to the total annual precipitation for Orbweaver2 hill3 slope profile. Output values are generated by the GP emulator designed by the first approach model.

model, we log-transformed daily precipiation amount for all 12 years. Number of periods in constructing basis for weight parameters is chosen as $q = 1$. To regularize the covariance matrix, we added *nugget* to the diagonal of Σ, as well. Thus the parameter $\theta = (\omega_1, \omega_2, \lambda)$ where $\omega_1 \in \mathbb{R}^{365}$ and $\omega_2 \in \mathbb{R}^{15}$ is transformed to

$$\theta = (\alpha_{10}, \alpha_{11}, \beta_{11}, \alpha_{20}, \alpha_{21}, \beta_{21}, \lambda)$$

Applying eBayes approach, we found the MLE estimate of theta:

$$\hat{\theta}_{MLE} = (-2.301884e + 02, -1.784269e + 02, 1.942628e + 02, -2.156876e + 02,$$

$$1.757213e + 02, -5.919064e + 01, 6.395477e - 04)$$

Estimated hyperparameters were plugged into the GP model. In Figures 4.6 and 4.7 we can see the soil loss distribution from data generated by WEPP simulation analysis and GP emulator for all 36 slope profiles in 2007 and 2018 years. GP emulator predicted almost the same amount of annual soil loss for all possible combinations of precipitation and slope profiles from the WEPP input data. Hence we attached results only for 2 years. The difference between emulated and WEPP outputted soil loss value has been attached in each figure. Note that plots of daily precipitation for years from 2007 to 2018 are given in the Figures A.1 and A.2 (Appendix 1).

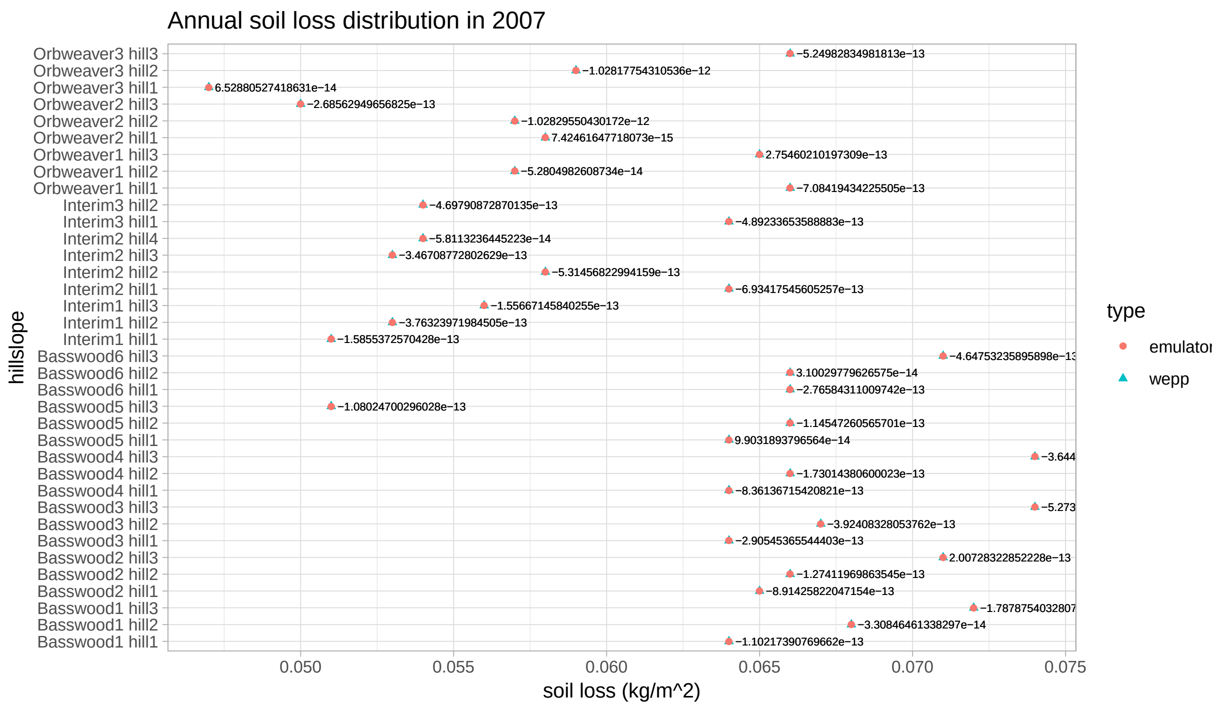

Figure 4.6 Soil loss distribution for all 36 slope profiles in the year 2007. The plot compares soil loss from WEPP simulation analysis and soil loss emulated by GP model. The difference between y and $\tilde{y}$ is displayed near each point.

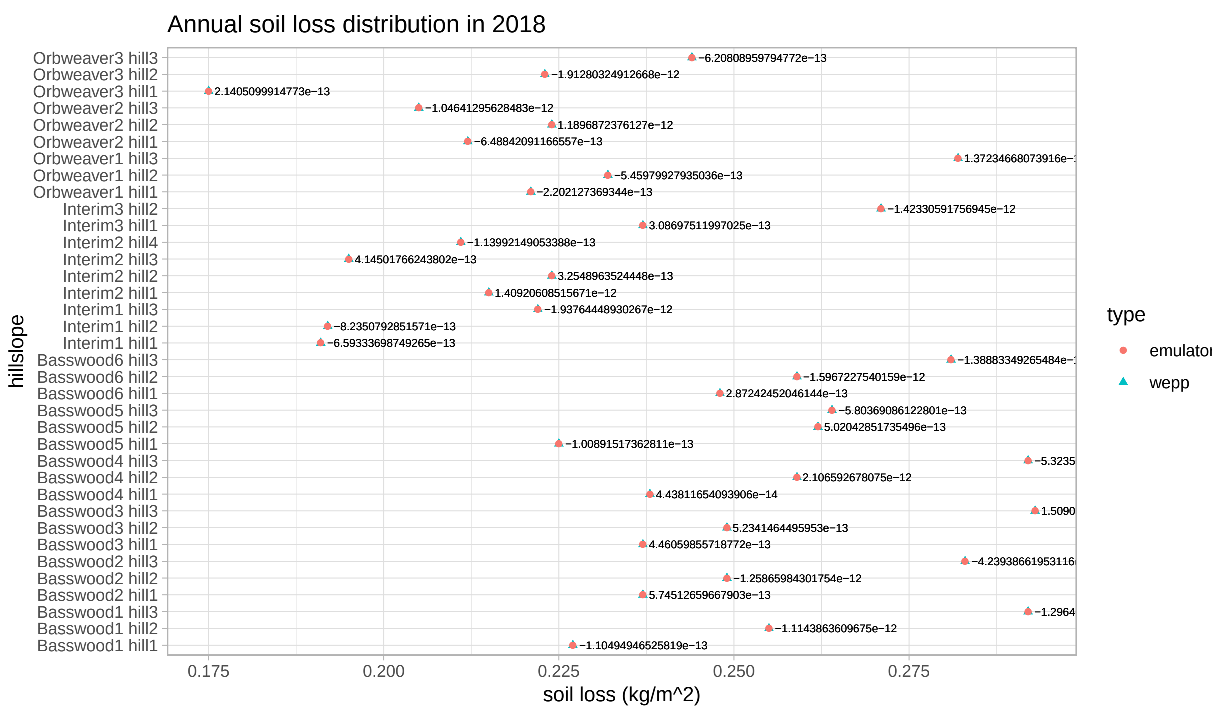

Figure 4.7 Soil loss distribution for all 36 slope profiles in the year 2018. The plot compares soil loss from WEPP simulation analysis and soil loss emulated by GP model. The difference between y and $\tilde{y}$ is displayed near each point.

CHAPTER 5. Discussion

We run three Gaussian process models to build an emulator for WEPP computer model. Only two covariates like yearly precipitation and slope profile of the hillslope were chosen as inputs due to their high correlation with the response - annual total soil loss amount.

5.1 GP emulator performance

For the simulation study, we managed to easily find the MLE estimates of the parameter $\theta = (\alpha_0, \alpha_1, \beta_1, \lambda)$. It is connected to the case that we already knew what are the real parameters and chose the initial values accordingly. Figure 4.1 suggests that the GP emulator passes almost through the same simulated output values: y. Since it's a simulation study, we already expect such a result. To see if the suggested GP emulator performs also well for the "real" data, we ran three separate GP models on the WEPP simulated data.

In the first model with only scalar inputs, we compare two heat maps. We see that spikes in annual soil loss amount from data generated by WEPP simulation analysis were smoothed out by GP emulator. This type of GP model ignores the functional structure of inputs, hence it assumes that different years with the same annual precipitation amount produce the same annual soil loss amount. Thus it's a weakness of the first model. In reality, the amount of soil loss depends on factors like temperature, wind direction and its velocity, and other extreme weather conditions. Assume *year 1* and *year 2* have the same annual precipitation amount. If in *year 1* the majority of precipitation was at below zero temperature, or the majority of precipitation took place in late summer, that year's soil loss amount will be different from the *year 2* where the majority of precipitation fall into spring period with above zero temperature. The same issue has to do with the average slope input since there we ignore the partial length steepness of the hill. If we add more scalar inputs (e.g., average max/min temperature, the average duration of precipitation, average wind velocity and etc) the simplest GP model is projected to perform better.

In the second approach, the GP model follows the y = WEPP simulated soil loss values. y' is within the 95% *confidence interval* of the predicted $\tilde{y}$ at a new locations $\tilde{x}$, which can be noted in Figures 4.4 and 4.5. Since the slope input is functional, we have predicted annual soil loss for specific hillslope profiles by varying the annual average precipitation.

Third attempt also shows that GP emulator passes almost at exactly the same y values and has a very small sum of squared residuals (SSR). Both inputs are functional, it's unclear how to order their values. Thus we predicted soil loss values only for existing WEPP precipitation and slope input values.

In general, all three models by the nature of Gaussian Processes smoothed out the spikes we saw in the WEPP simulated data. As expected the third model which has only functional inputs showed the best performance. For instance, functional inputs as daily precipitation amount, apparently, carry more information than scalar annual precipitation amount. The only drawback of the third model is the increasing complexity in calculating and getting inverse of the covariance matrix.

5.2 Next steps

There are many possible ways of sophisticating GP emulator. From the model construction and data informativity perspective we can add more input variables to the GP model. For sure, features from the WEPP climate input file like duration of precipitation, daily min/max temperature, wind direction, wind velocity have an impact on the soil erosion. Tillage and crop characteristics, also the texture of the soil itself have a prominent impact on the amount of soil erosion as well. We will need to carry out a new design for adding those inputs as scalar/functional numerical inputs. Daily precipitation amount has a natural functional design, while functional slope input was designed manually. Likewise, slope measurements were taken from "face" transect and there is another type of transect - "gulley". For instance, we can conduct a sensitivity analysis of the GP model on the transect type. Categorical variables as drought, flood, storm and etc. can be also added to the mean function $\mu(x)$.

To decrease the time required for the estimation of hyperparameters, we can include the analytical gradient function of the likelihood function to the $optim()$. Choosing better initial parameters would also speed up the performance. For instance, weight parameters presumably have a right skewed, heavy-tailed distribution which is mostly concentrated at 0.

The most important known drawback of GP is the computation time. Large sample size, hence large training dataset only increases the dimension of the Σ covariance matrix, thus slows down the maximum likelihood method. Optimization algorithm needs to calculate Σ^{-1} at each parameters update, where the complexity of such operation is $O(n^3)$. A possible remedy would be to replace direct inversion of a matrix with another procedure which will cost less, possibly $O(n^2)$.

Assuming the method we developed in this research, we would like to turn our attention to its third type: GP model with functional inputs. This model does not take into account the interactions between inputs and it also ignores the spatial characteristics of inputs. As a next step, we would like to make this GP model more scalable and applicable for space-varying functional inputs too. Our research team at C-CHANGE now is working on constructing a GP emulator for another computer model - APSIM, which takes in 12 functional, both time- and space- varying inputs.

APPENDIX A. Appendix 1

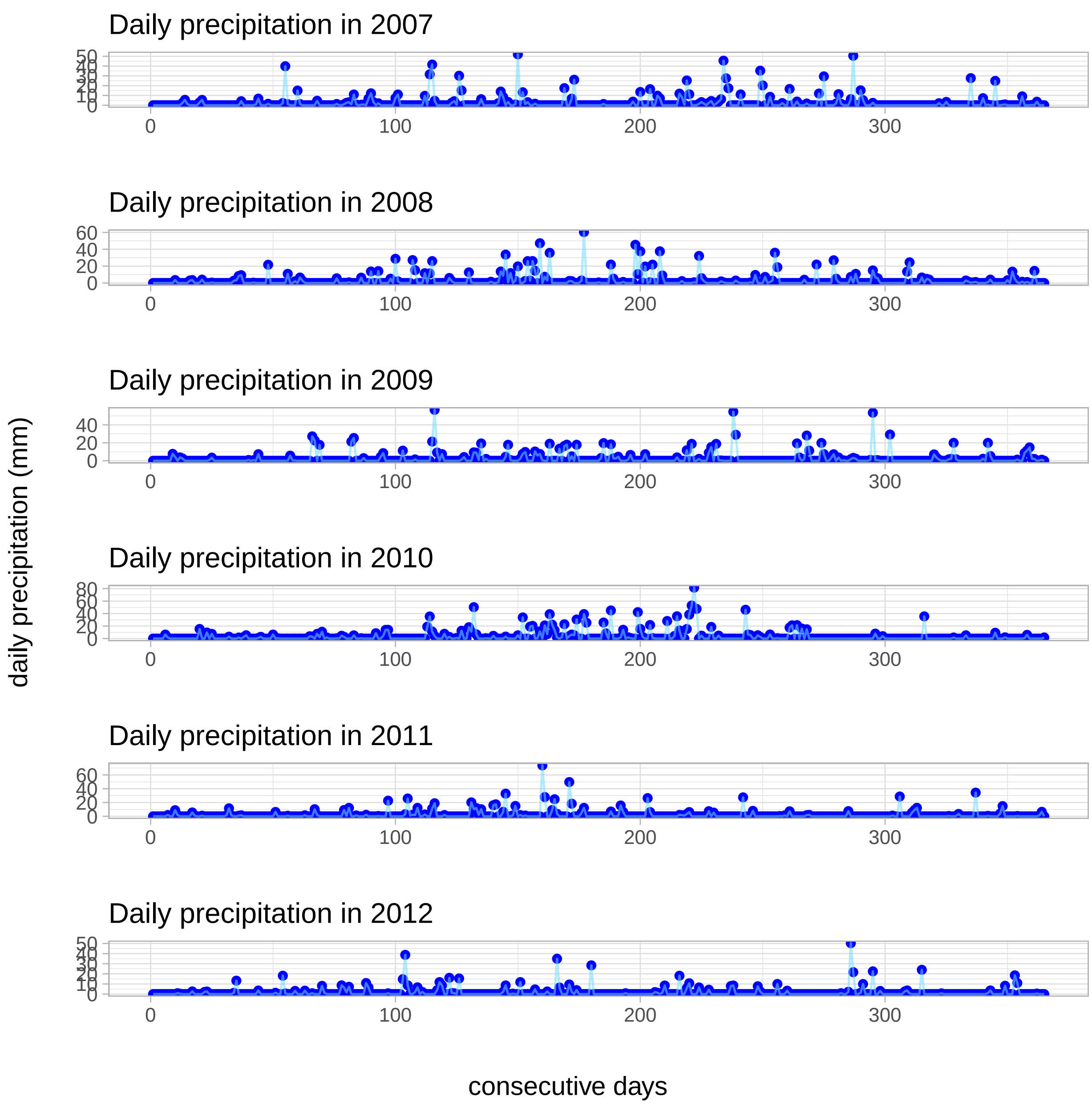

Figure A.1 Daily precipitation amount from 2007 to the 2012 year. Data is collected from the NSNWR neighborhood climate station.

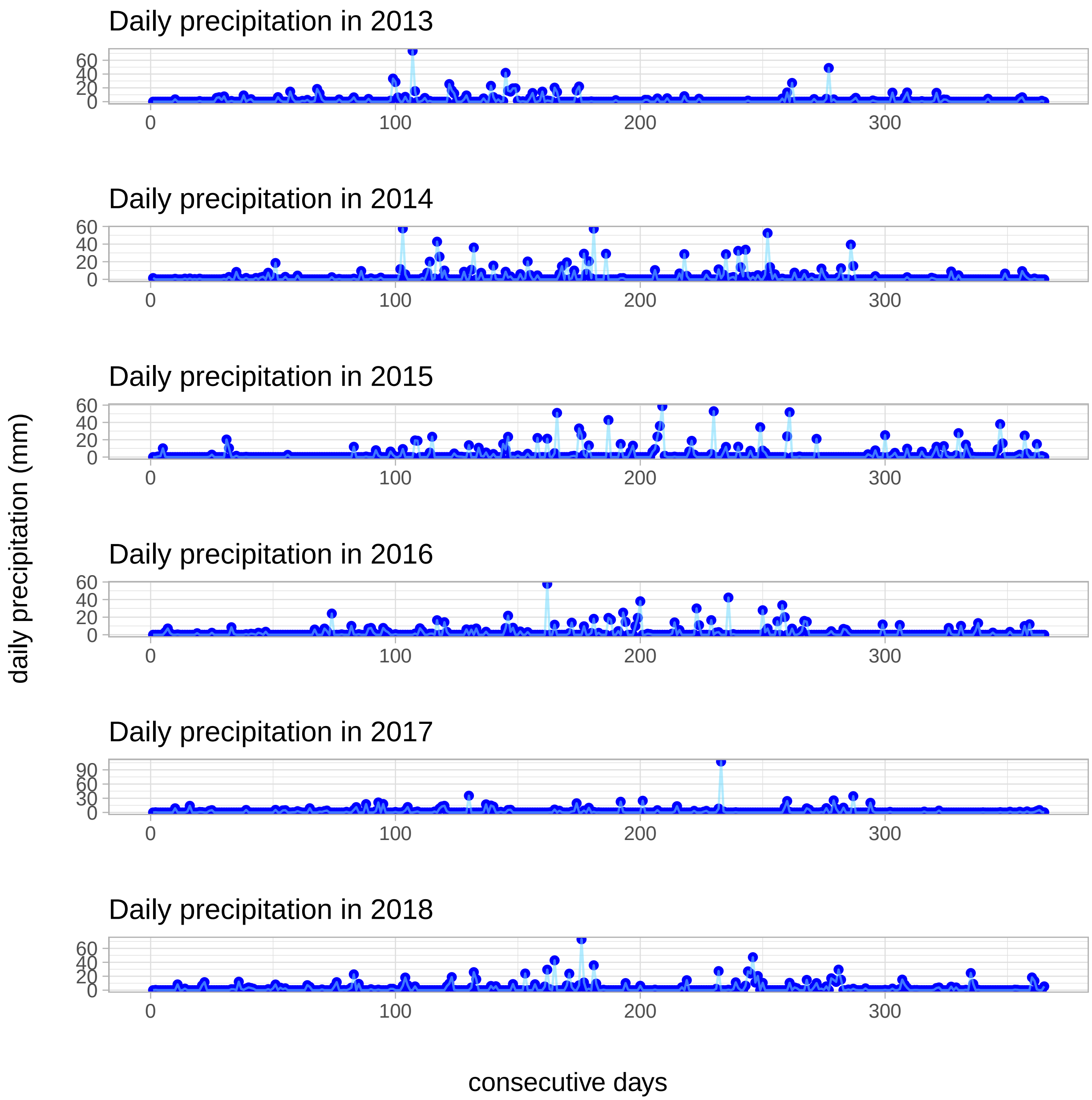

Figure A.2 Daily precipitation amount from 2013 to the 2018 year. Data is collected from the NSNWR neighborhood climate station.

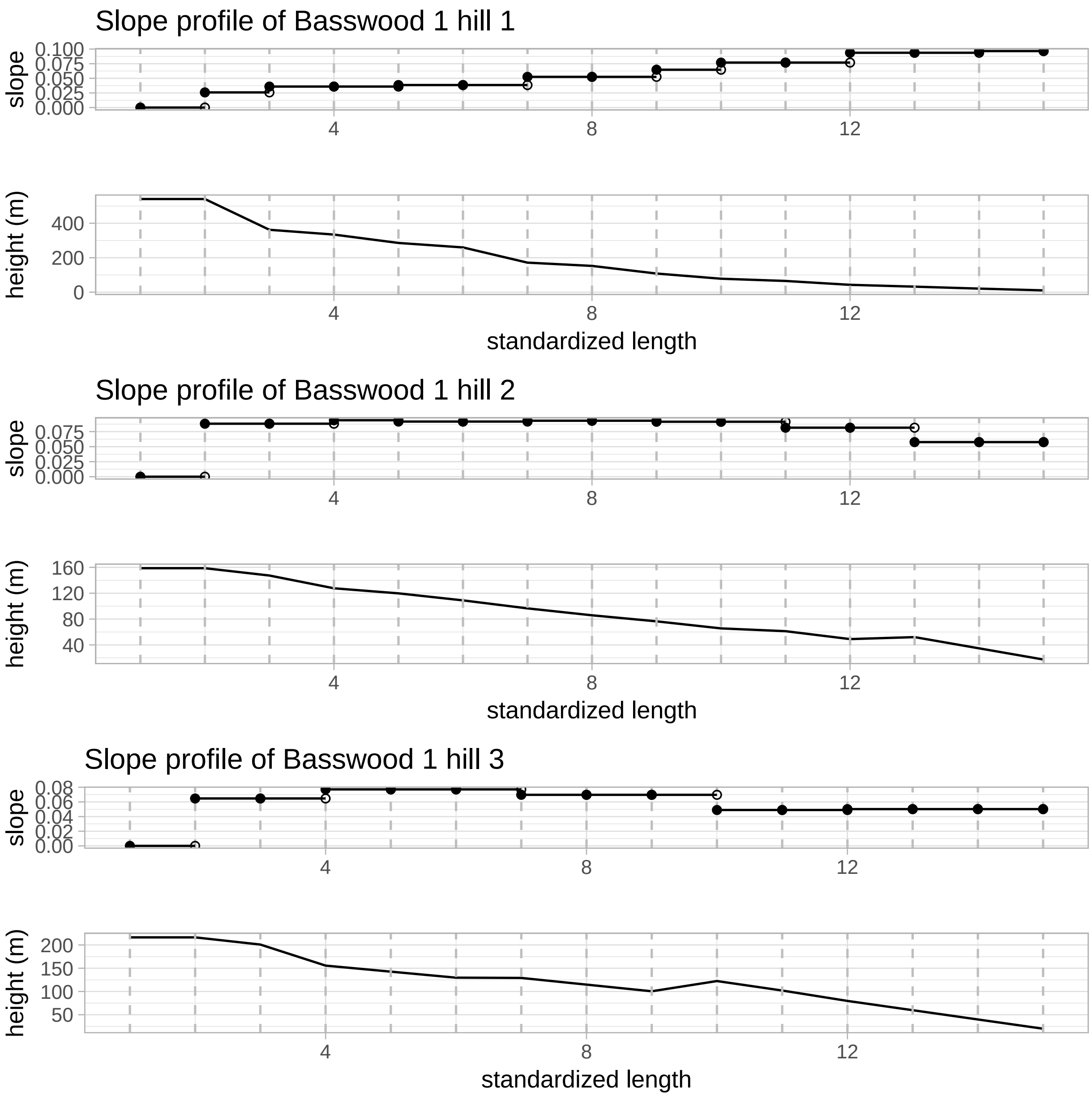

Figure A.3 Slope profile of 3 hillslopes in Basswood1 agricultural watershed at NSNWR.

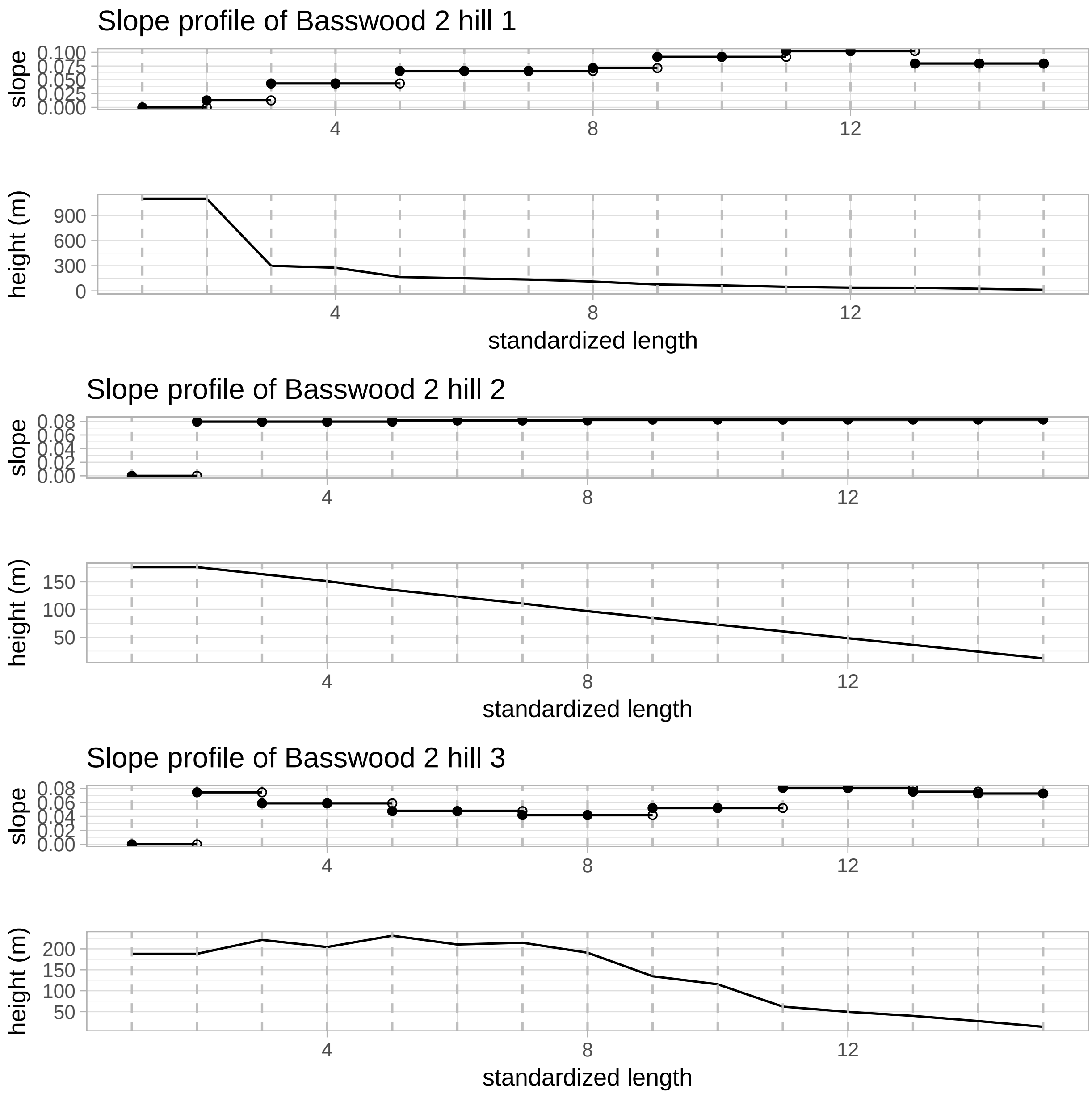

Figure A.4 Slope profile of 3 hillslopes in Basswood2 agricultural watershed at NSNWR.

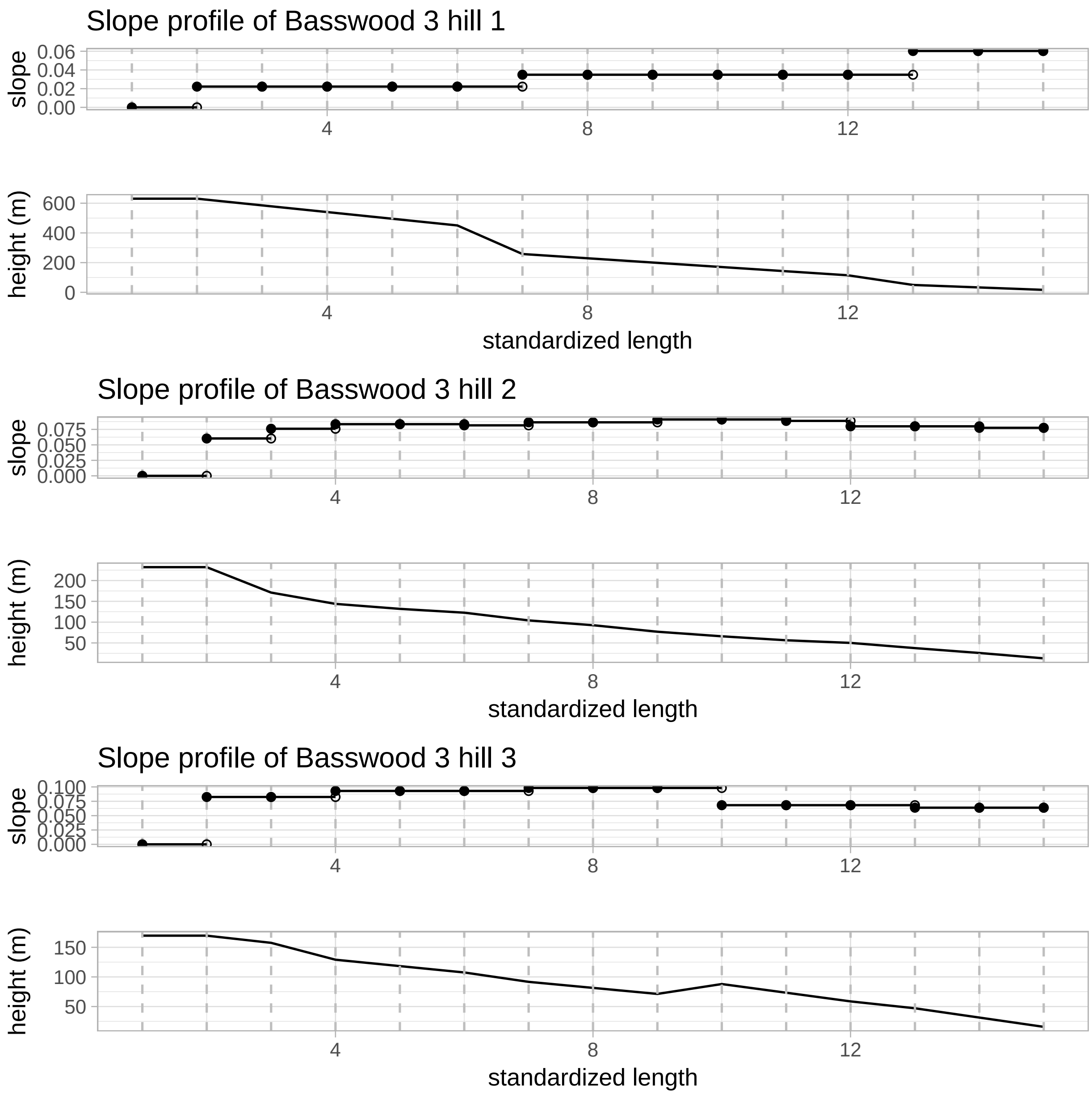

Figure A.5 Slope profile of 3 hillslopes in Basswood3 agricultural watershed at NSNWR.

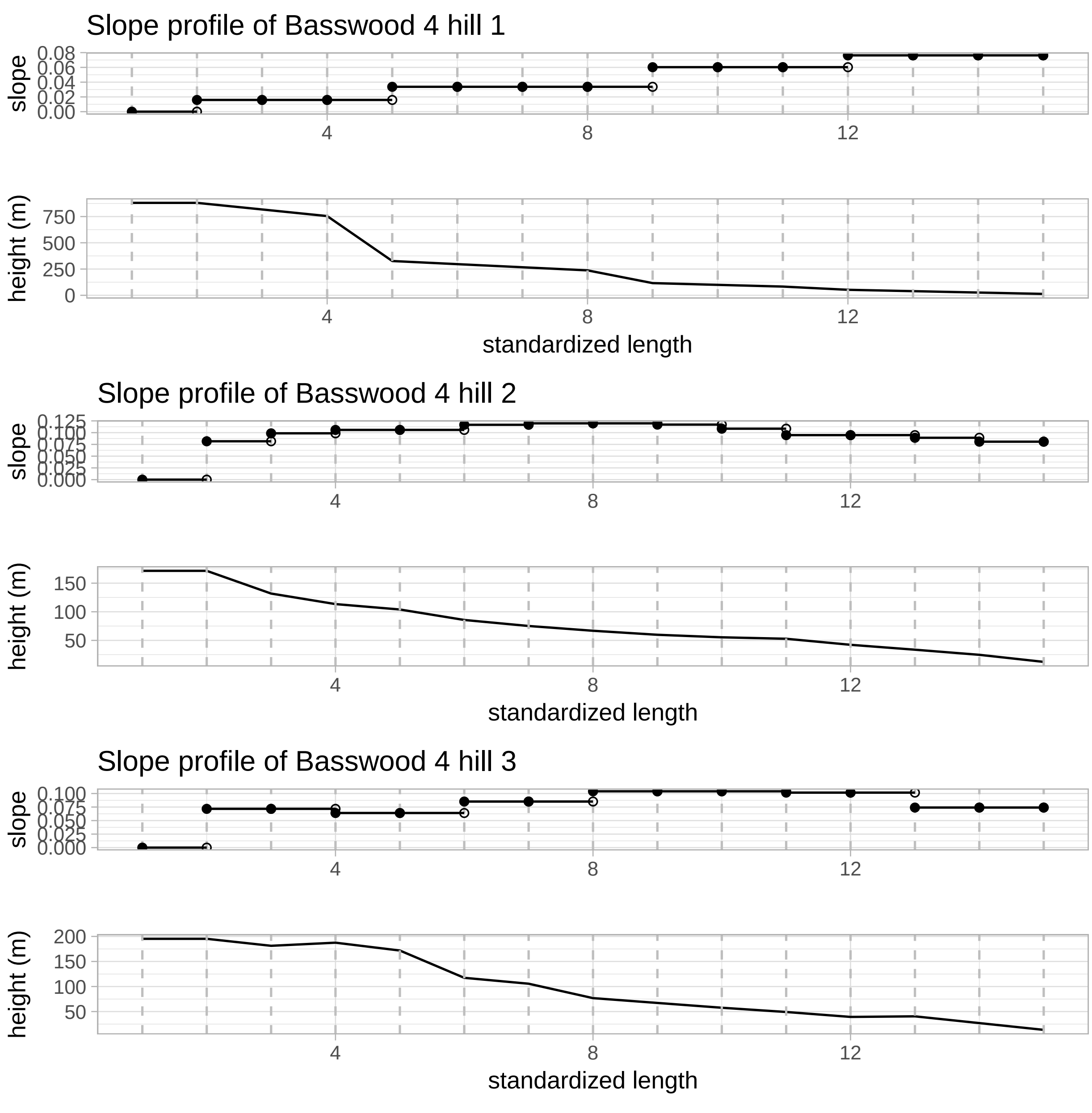

Figure A.6 Slope profile of 3 hillslopes in Basswood4 agricultural watershed at NSNWR.

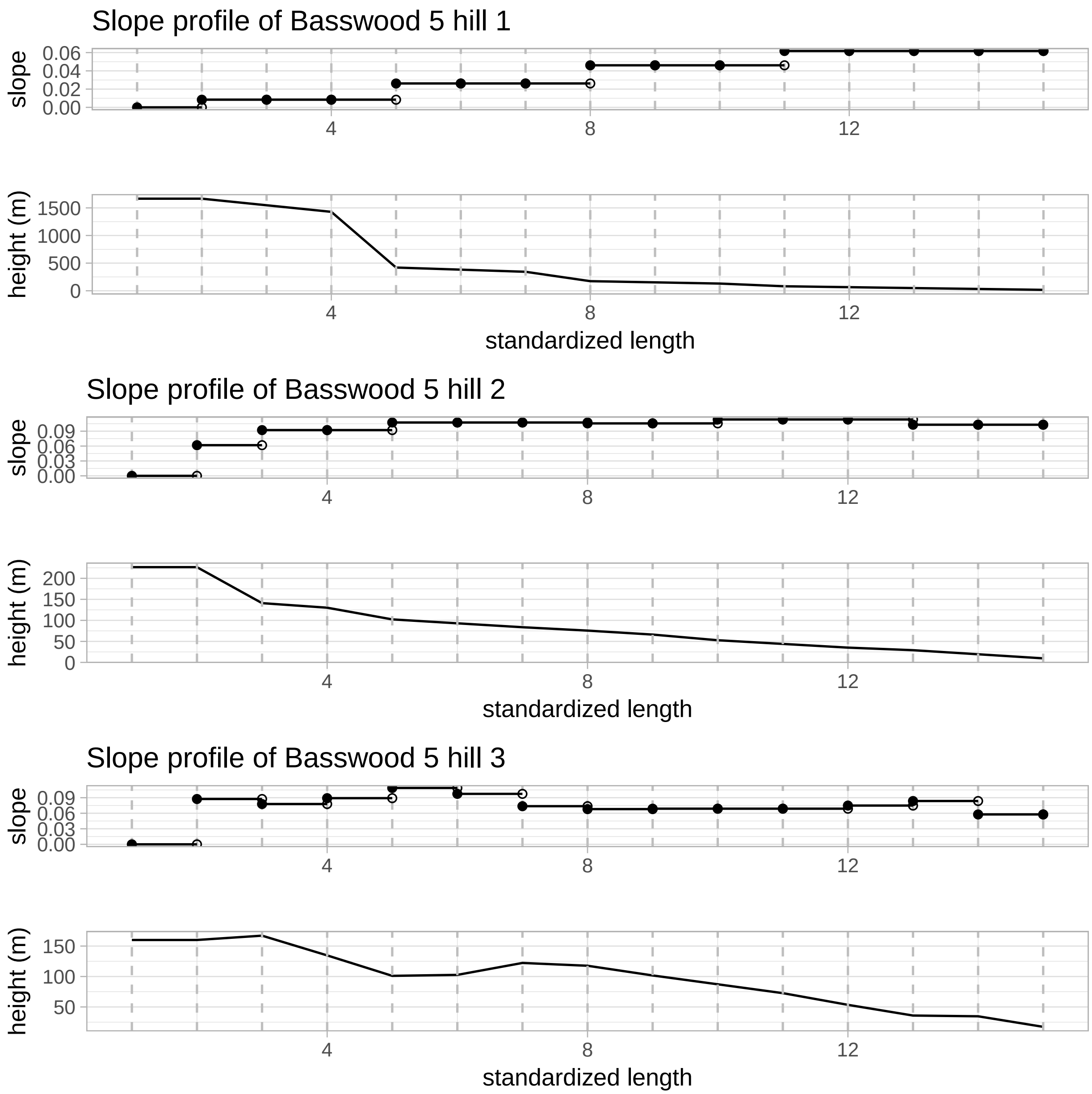

Figure A.7 Slope profile of 3 hillslopes in Basswood5 agricultural watershed at NSNWR.

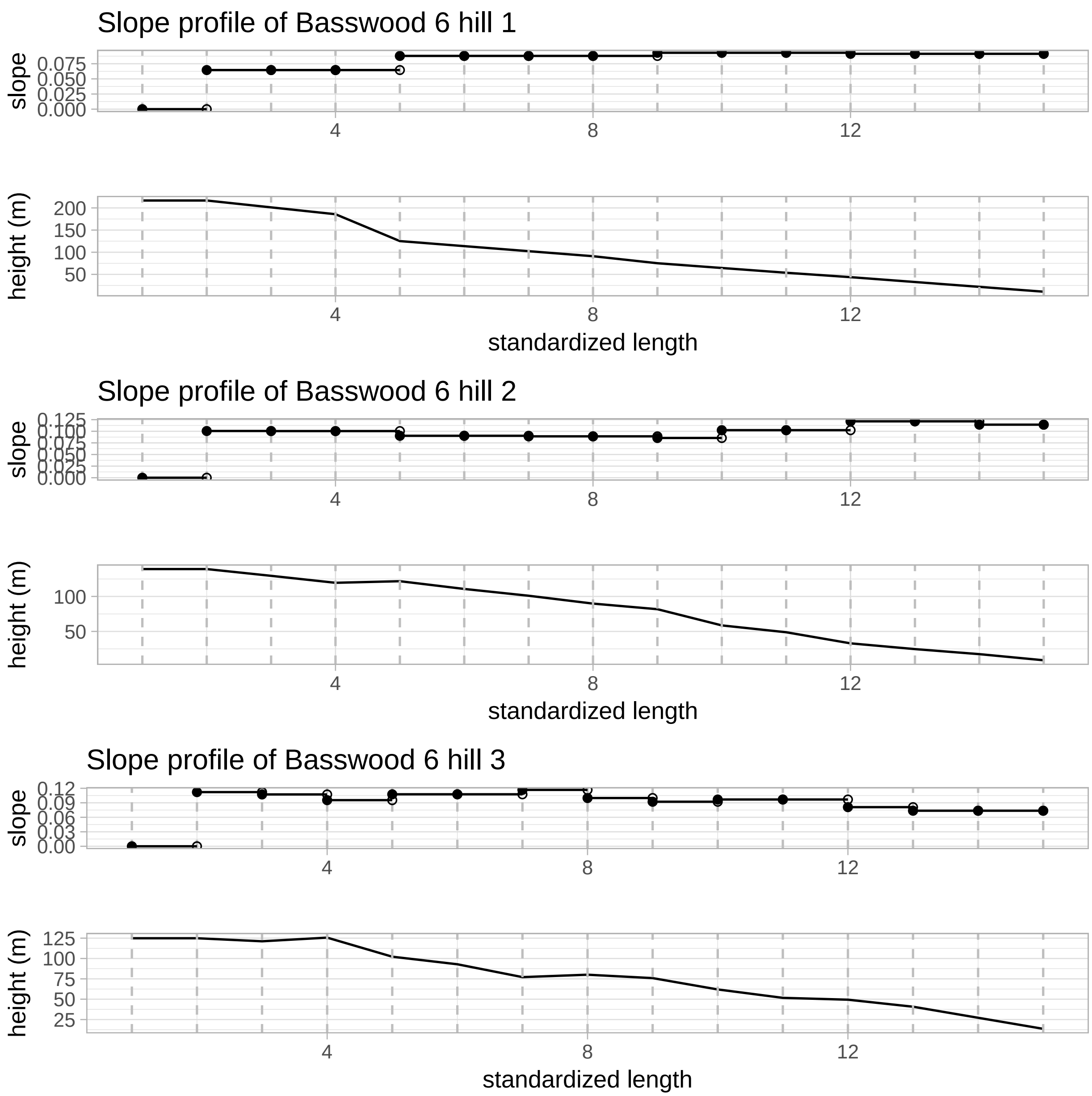

Figure A.8 Slope profile of 3 hillslopes in Basswood6 agricultural watershed at NSNWR.

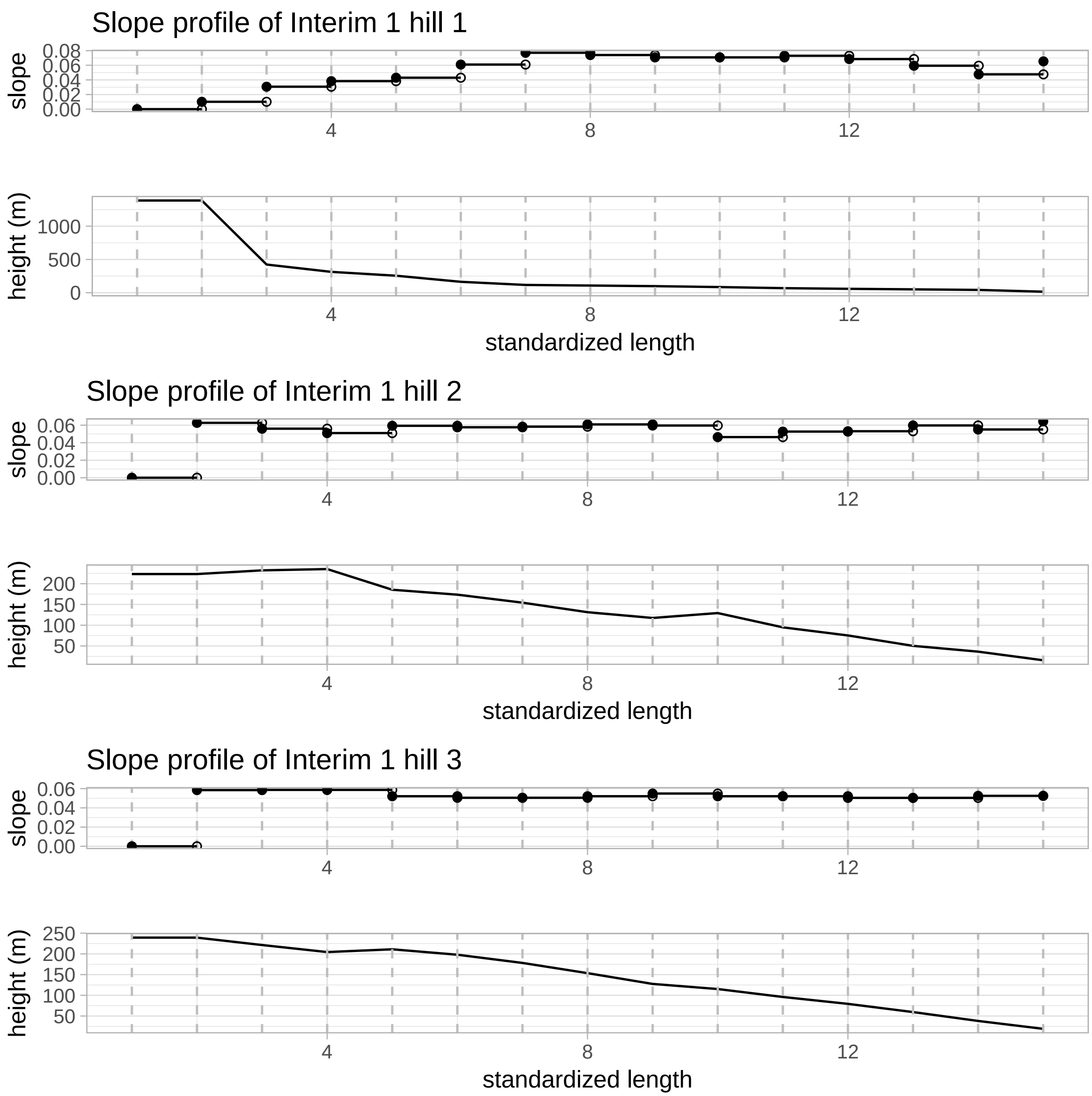

Figure A.9 Slope profile of 3 hillslopes in Interim1 agricultural watershed at NSNWR.

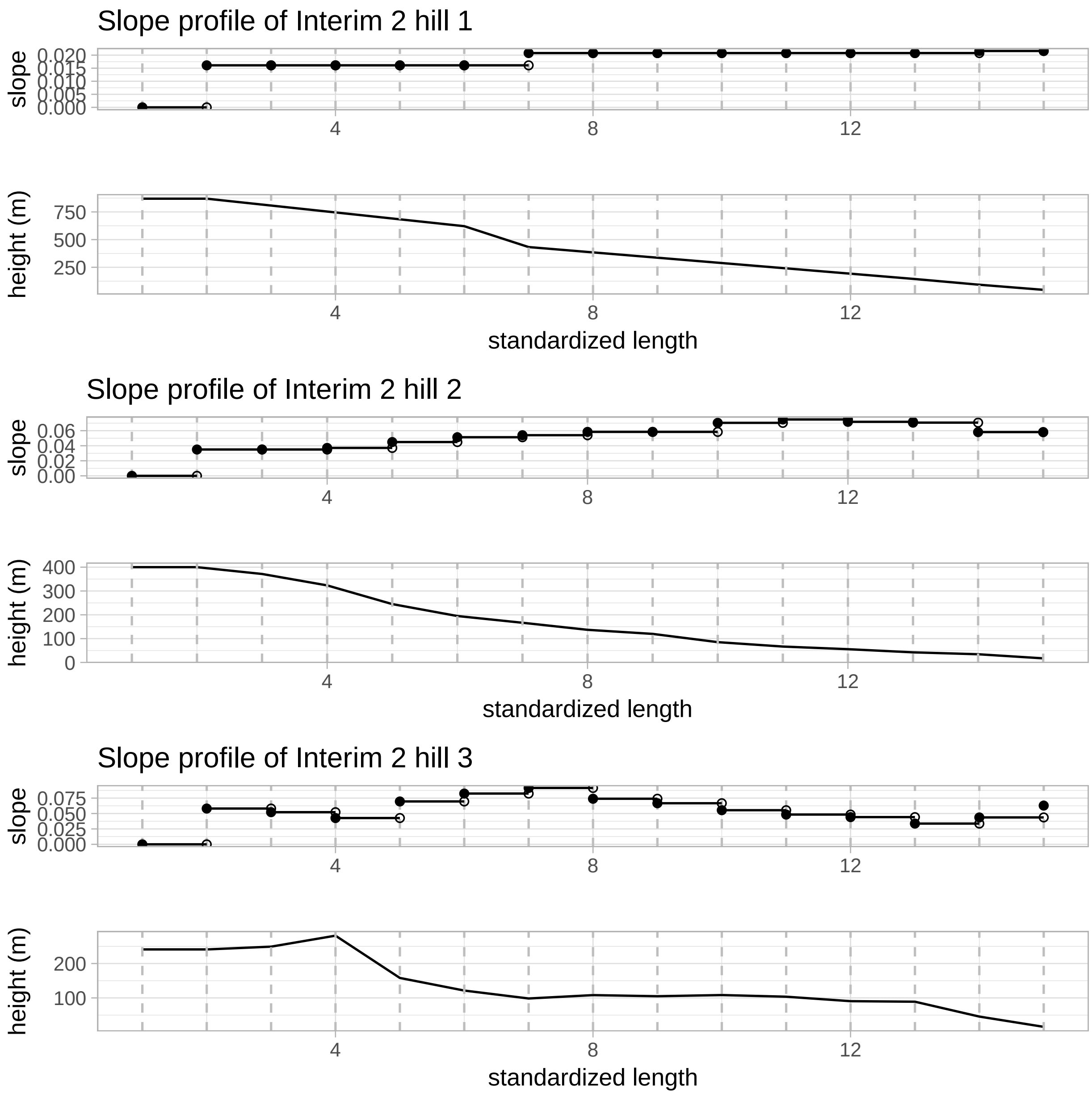

Figure A.10 Slope profile of 3 hillslopes in Interim2 agricultural watershed at NSNWR.

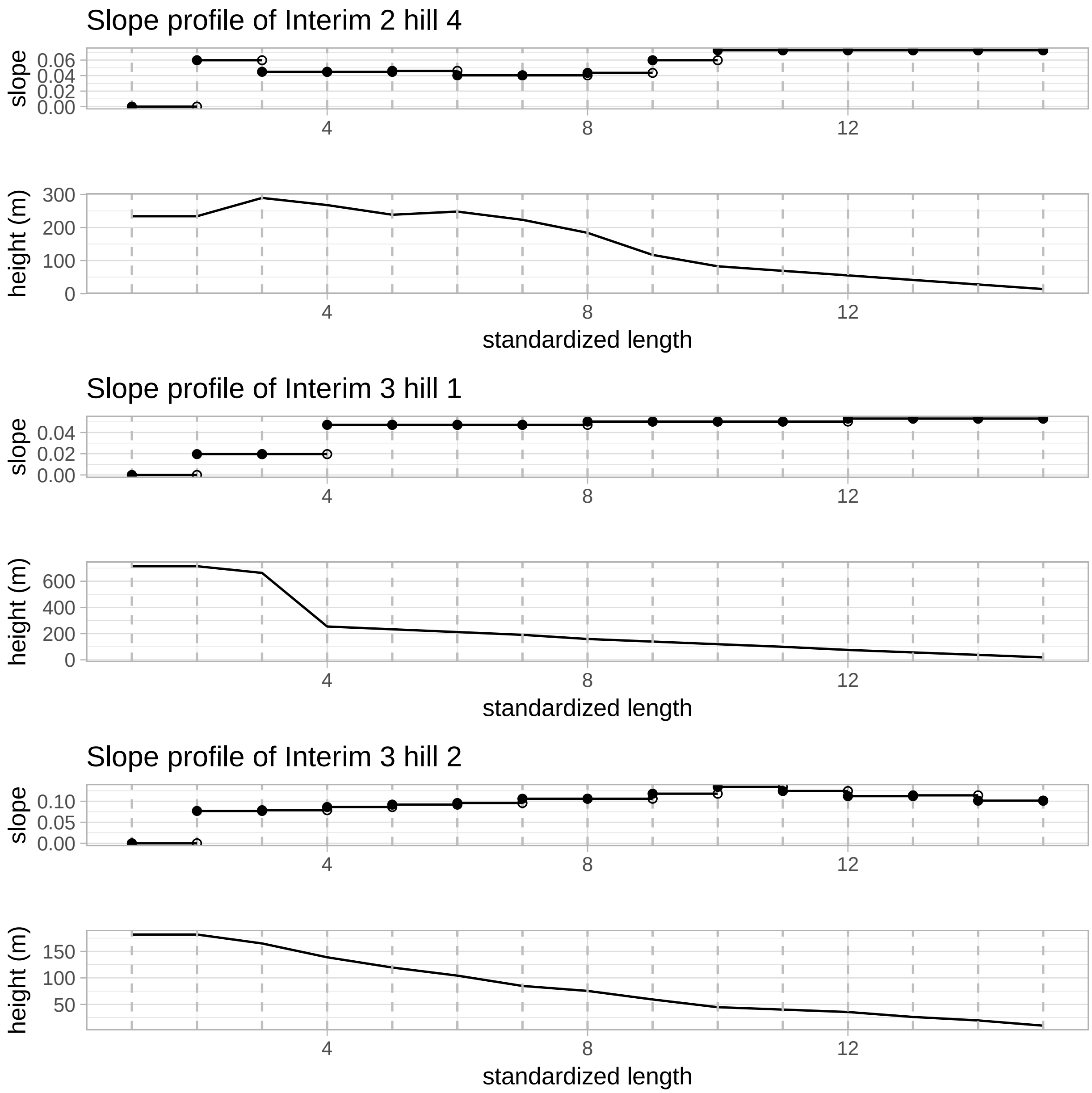

Figure A.11 Slope profile of 1 hillslope in Interim2 and 2 hillslopes in Interim 3 agricultural watershed at NSNWR.

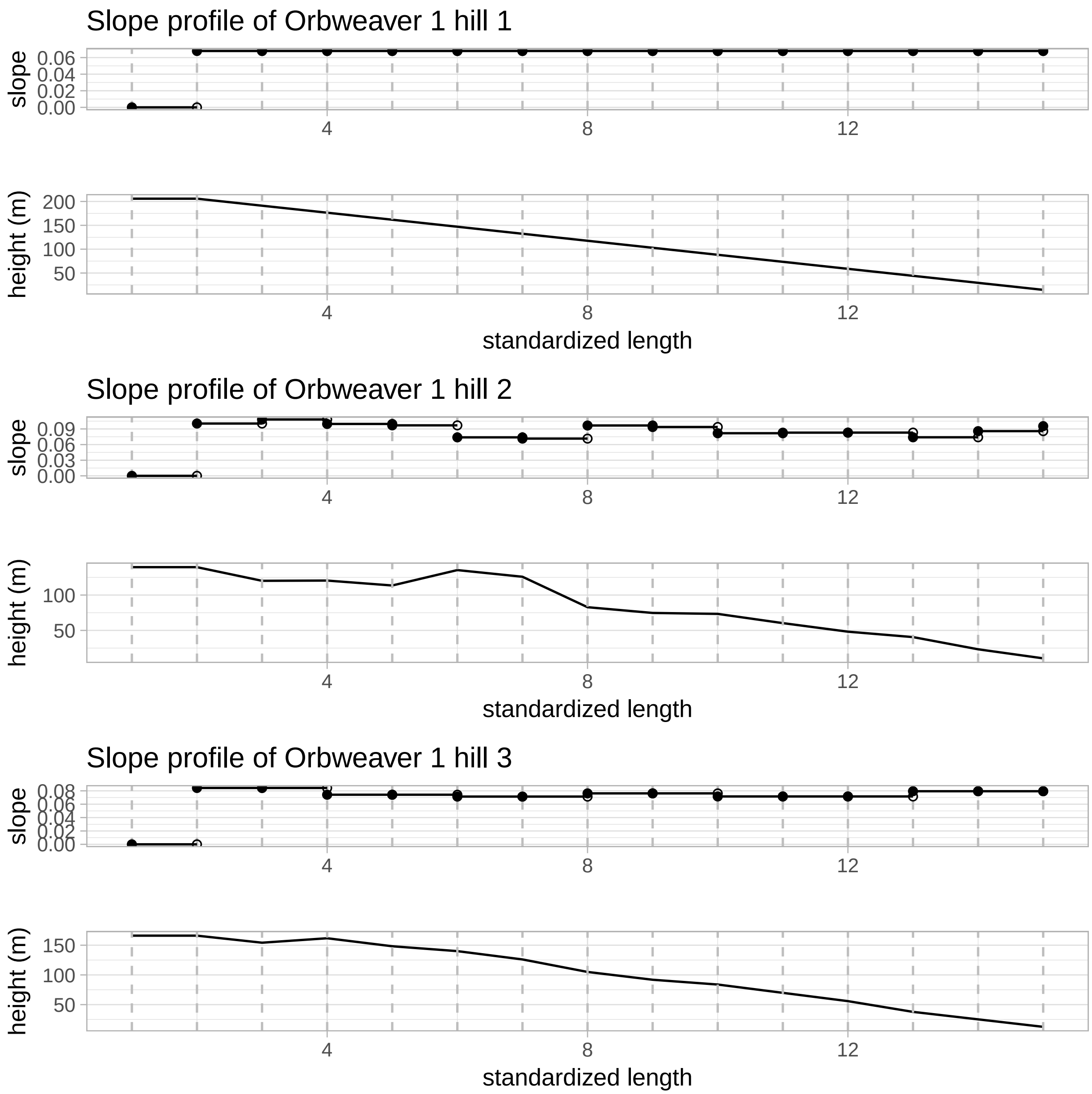

Figure A.12 Slope profile of 3 hillslopes in Orbweaver1 agricultural watershed at NSNWR.

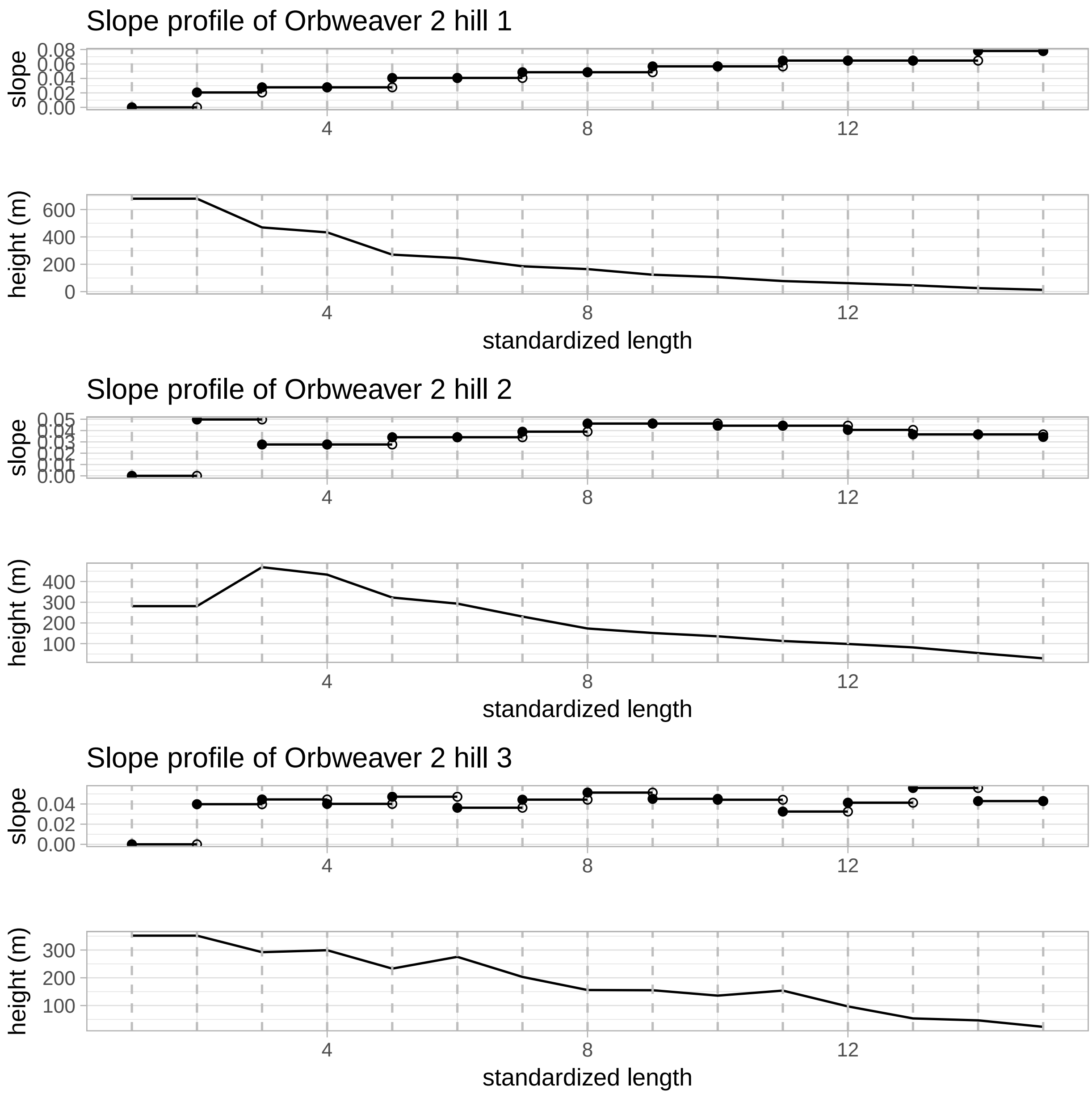

Figure A.13 Slope profile of 3 hillslopes in Orbweaver2 agricultural watershed at NSNWR.

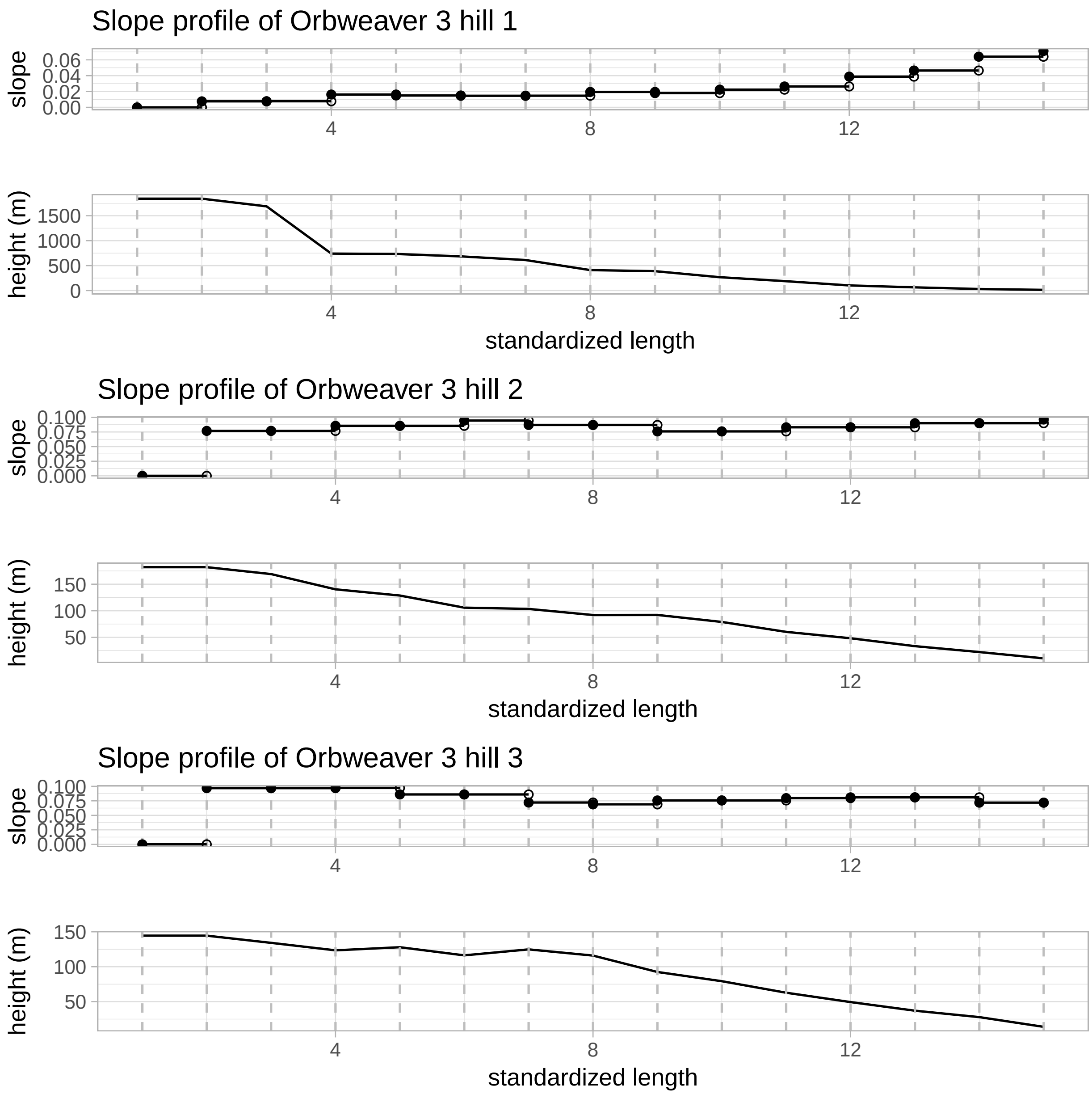

Figure A.14 Slope profile of 3 hillslopes in Orbweaver3 agricultural watershed at NSNWR.

BIBLIOGRAPHY

Eller, D. (2014). Erosion estimated to cost iowa $1 billion in yield. *Des Moines Register*.

Flanagan, D. C., Gilley, J. E., and Franti, T. G. (2007). Water erosion prediction project (wepp): Development history, model capabilities, and future enhancements. *Transactions of the ASABE*, 50(5):1603–1612.

Morris, M. D. (2014). Maximin distance optimal designs for computer experiments with time-varying inputs and outputs. *Journal of statistical Planning and Inference*, 144:63–68.

NRCS (2010). Soil survey staff, natural resources conservation service, united states department of agriculture. *Agricultural and Forest Meteorology*.

Petris, G., Petrone, S., and Campagnoli, P. (2009). Dynamic linear models. In *Dynamic Linear Models with R*, pages 31–84. Springer.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, pages 409–423.

STYLE, F. (2005). Natural resources conservation service conservation practice standard.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA.

Zhou, X., Helmers, M. J., Asbjornsen, H., Kolka, R., and Tomer, M. D. (2010). Perennial filter strips reduce nitrate levels in soil and shallow groundwater after grassland-to-cropland conversion. *Journal of environmental quality*, 39(6):2006–2015.